(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification**[7]: **G06F 17/30**

(21) **International Application Number:** PCT/US03/15638

(22) **International Filing Date:** 15 May 2003 (15.05.2003)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
10/153,151   21 May 2002 (21.05.2002)   US

(63) **Related by continuation (CON) or continuation-in-part (CIP) to earlier application:**
US   10/153,151 (CON)
Filed on   21 May 2002 (21.05.2002)

(71) **Applicant** (for all designated States except US): **WASHINGTON UNIVERSITY** [US/US]; One Brookings Drive, St. Louis, MO 63130 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** (for US only): **INDECK, Ronald,**

S. [US/US]; 729 Gralee, St. Louis, MO 63132 (US). **CYTRON, Ron, Kaplan** [US/US]; 49 Granada Way, St. Louis, MO 63124 (US). **FRANKLIN, Mark, Allen** [US/US]; 7456 Stratford Avenue, St. Louis, MO 63130 (US). **CHAMBERLAIN, Roger, D.** [US/US]; 10920 Leighton Court, St. Louis, MO 63146 (US).

(74) **Agents: HAFERKAMP, Richard, E.** et al.; Thompson Coburn LLP, One US Bank Plaza, St. Louis, MO 63101 (US).

(81) **Designated States** (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),

(54) **Title:** ASSOCIATIVE DATABASE SCANNING AND INFORMATION RETRIEVAL USING FPGA DEVICES
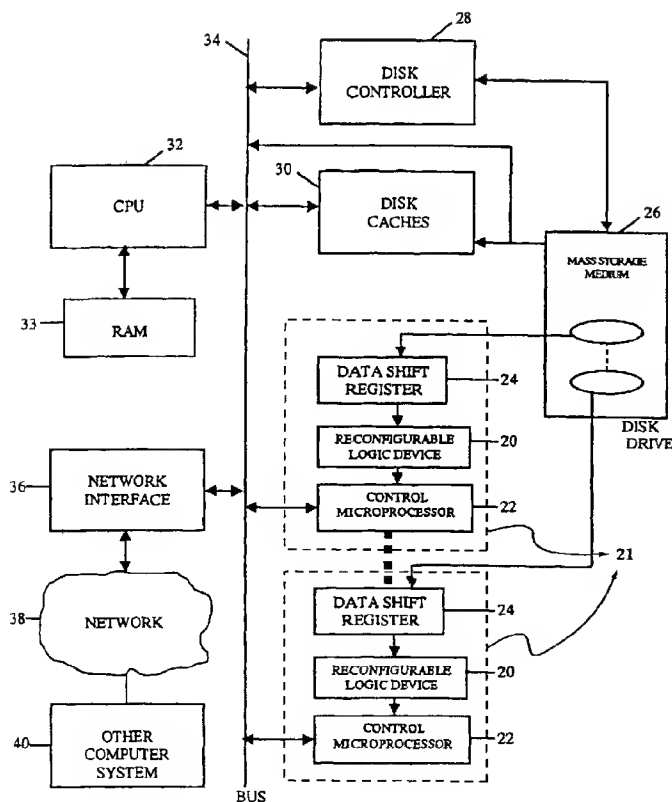
(57) **Abstract:** A data storage and retrieval device and method is disclosed. The device includes at least one magnetic storage medium configured to store target data and at least one re-configurable logic device comprising an FPGA coupled to the at least one magnetic storage medium and configured to read a continuous stream of target data therefrom, having been configured with a template or as otherwise desired to fit the type of search and data being searched. The re-configurable logic device is configured to receive at least one search inquiry in the form of a data key and to determine a match between the data key and the target data as it is being read from the at least one magnetic storage medium. This device and method can perform a variety of searches on the target data including without limitation exact and approximate match searches, sequence match searches, image match searches and data reduction searches. This device and method may be provided as part of a stand-alone computer system, embodied in a network attached storage device, or can otherwise be provided as part of a computer LAN or WAN.

Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

ASSOCIATIVE DATABASE SCANNING AND INFORMATION RETRIEVAL USING FPGA DEVICES

Cross Reference to Related Application:

This application is a continuation-in-part of serial no. 09/545,472, the disclosure of which is incorporated herein by
5   reference.

Background of the Invention

Indications are that the average database size and associated
10   software support systems are growing at rates that are greater than the increase in processor performance (i.e., more than doubling roughly every 18 months). This is due to a number of factors including without limitation the desire to store more detailed information, to store information over longer periods of time, to
15   merge databases from disparate organizations, and to deal with the large new databases which have arisen from emerging and important applications. For example, two emerging applications having large and rapidly growing databases are those connected with the genetics revolution and those associated with cataloging and accessing
20   information on the Internet. In the case of the Internet, current industry estimates are that in excess of 1.5 million pages are added to the Internet each day. At the physical level this has been made

2

possible by the remarkable growth in disk storage performance where magnetic storage density has been doubling every year or so for the past five years.

Search and retrieval functions are more easily performed on
5   information when it is indexed. For example, with respect to financial information, it can be indexed by company name, stock symbol and price. Oftentimes, however, the information being searched is of a type that is either hard to categorize or index or which falls into multiple categories. As a result, the accuracy of a
10   search for information is only as good as the accuracy and comprehensiveness of the index created therefor. In the case of the Internet, however, the information is not indexed. The bottleneck for indexing is the time taken to develop the reverse index needed to access web pages in reasonable time. For example, while there are
15   search engines available, designing a search which will yield a manageable result is becoming increasingly difficult due to the large number of "hits" generated by less than a very detailed set of search instructions. For this reason, several "intelligent" search engines have been offered on the web, such as Google, which are intended to
20   whittle down the search result using logic to eliminate presumed undesired "hits".

With the next-generation Internet, ever-faster networks, and expansion of the Internet content, this bottleneck is becoming a critical concern. Further, it is becomingly exceedingly difficult to
25   index information on a timely basis. In the case of the Internet, current industry estimates are that in excess of 1.5 million pages are added to the Internet each day. As a result, maintaining and updating a reverse index has become an enormous and continuous task and the bottleneck it causes is becoming a major impediment to the
30   speed and accuracy of existing search and retrieval systems. Given the ever increasing amounts of information available, however, the ability to accurately and quickly search and retrieve desired information has become critical.

Associative memory devices for dealing with large databases are
35   known in the prior art. Generally, these associative memory devices comprise peripheral memories for computers, computer networks, and the like, which operate asynchronously to the computer, network, etc. and provide increased efficiency for specialized searches. Additionally, it is also known in the prior art that these memory

devices can include certain limited decision-making logic as an aid to a main CPU in accessing the peripheral memory. An example of such an associative memory device particularly adapted for use with a rotating memory such as a high speed disk or drum can be found in

5   U.S. Patent No. 3,906,455, the disclosure of which is incorporated herein by reference. This particular device provides a scheme for use with a rotating memory and teaches that two passes over a memory sector is necessary to presort and then sort the memory prior to performing any logical operations thereon. Thus, this device is

10 taught as not being suitable for use with any linear or serial memory such as magnetic tape or the like.

Other examples of prior art devices may also be found in U.S. Patent Nos. 3,729,712; 4,464,718; 5,050,075; 5,140,692; and 5,721,898; the disclosures of which are incorporated herein by

15 reference.

As an example, in 4,464,718, Dixon performs fixed comparisons on a fixed number of bytes. They don't have the ability to scan and correlate arbitrarily over the data. They search serially along the tracks in a given disk cylinder but there is no provision for

20 parallel searching across disks. Dixon's comparisons are limited by a fixed rigid number of standard logical operation types. Additionally, the circuitry presented supports only these single logical operations. There is no support for approximate or fuzzy matching.

25 While these prior art associative memory devices represent an attempt to speed the input and output of information to and from a peripheral memory, which in many cases is a mass storage memory device, all rely on the classic accessing of data stored in digital form by reading and interpreting the digital either address or

30 content of the memory location. In other words, most such devices access data by its address but there are some devices that take advantage of the power of content addressing as is well known in the art. Nevertheless, in all of the prior art known to the inventors, the digital value of the address or data contained in the addressed

35 location must be read and interpreted in its digital form in order to identify the data and then select it for processing. Not only does it take processing time to read and interpret the digital data represented by the address or content, this necessarily requires that the accessing circuit process the memory according to the structure

4

of the data stored. In other words, if the data is stored in octets,
then the accessing circuitry must access the data in octets and
process it in an incremental manner. This "start and stop"
processing serves to increase the input/output time required to
5   access data. As is also well known in the art, this input/output
time typically represents the bottleneck and effective limitation of
processing power in any computer or computer network.

Furthermore, given the vast amount of information available to
be searched, data reduction operations (i.e., the ability to
10 summarize data in some aggregate form) has become critical.
Oftentimes, the ability to quickly perform data reduction functions
can provide a company with a significant competitive advantage.

Likewise, with the improvements in digital imaging technology,
the ability to perform two dimensional matching such as on images has
15 become necessary. For example, the ability to conduct matches on a
particular image of an individual, such as his or her face or retina,
or on a fingerprint, is becoming critical to law enforcement as it
steps up its efforts on security in light of the September 11, 2001
terrorist attacks. Image matching is also of importance to the
20 military in the area of automatic target recognition.

Finally, existing searching devices cannot currently be quickly
and easily reconfigured in response to changing application demands.

Accordingly, there is a need for an improved information search
and retrieval system and method which overcomes these and other
25 problems in the prior art.

In order to solve these and other problems in the prior art,
the inventors herein have succeeded in designing and developing a
method and apparatus for an associative memory using Field
Programmable Gate Arrays (FPGA) in several embodiments which provide
30 an elegantly simple solution to these prior art limitations as well
as dramatically decreased access times for data stored in mass
storage memories. As described below, the invention has several
embodiments each of which has its own advantages.

The parent of the present invention discloses and claims the
35 use of programmable logic and circuitry generally without being
specific as to any choice between the various kinds of devices
available for this part of the invention. In this patent filing, the
inventors are disclosing more specifically the use of FPGA's as part
of the circuitry for various reasons as their best mode. There are

5

several reasons for that. The first of these is speed. And, there are two different aspects of operation in which speed plays a part. The first of these is the speed of reconfiguration. It is known in the art that FPGA's may be quickly programmed in the field to

5   optimize the search methodology using a template, the template having been prepared in advance and merely communicated to the FPGA's over a connecting bus. Should it then be desired to search using a different methodology, the FPGA's may then be quickly and conveniently re-programmed with another prepared template in a

10   minimal number of clock cycles and the second search started immediately. Thus, with FPGA's as the re-configurable logic, shifting from one search to another is quite easy and quick, relative to other types of re-programmable logic devices.

A second aspect of speed is the amount of time required, once

15   programmed, a search requires. As FPGA's are hardware devices, searching is done at hardware processing speeds which is orders of magnitude faster than at software processing speeds as would be experienced with a microprocessor, for example. Thus, FPGA's are desirable over other software implementations where speed is a

20   consideration as it most often is.

In considering the use of templates, it is contemplated that at least several "generic" templates would be prepared in advance and would be available for use in performing text searching in either an absolute search, an approximate search, or a higher or advanced

25   search mode incorporating a Boolean algebra logic capability, or a graphics search mode. These could then be stored in a CPU memory and be available either on command or loaded in automatically in response to a software queue indicating one of these searches.

Still another factor to consider is cost, and the recent price

30   reductions in FPGA's have made them more feasible for implementation as a preferred embodiment for this application, especially as part of a hard disk drive accelerator as would be targeted for a pc market. It is fully expected that further cost reductions will add to the desirability of these for this implementation, as well as others as

35   discussed in greater detail below.

Generally, the invention may be described as a technique for data retrieval through approximate matching of a data key with a continuous reading of data as stored on a mass storage medium, using FPGA's to contain the template for the search and do the comparison,

6

all in hardware and at essentially line speed. By utilizing FPGA's, the many advantages and features commonly known are made available. These include the ability to arrange the FPGA's in a "pipeline" orientation, in a "parallel" orientation, or even in an array

5   incorporating a complex web overlay of interconnecting data paths allowing for complex searching algorithms. In its broadest, and perhaps most powerful, embodiment, the data key may be an analog signal and it is matched with an analog signal generated by a typical read/write device as it slews across the mass storage medium. In

10   other words, the steps taught to be required in the prior art of not only reading the analog representation of digital data stored on the mass storage medium but also the conversion of that signal to its digital format prior to being compared are eliminated. Furthermore, there is no requirement that the data be "framed" or compared

15   utilizing the structure or format in which the data has been organized and stored. For an analog signal, all that need be specified is the elapsed time of that signal which is used for comparison with a corresponding and continuously changing selected time portion of the "read" signal. Using any one of many standard

20   correlation techniques as known in the prior art, the data "key" may then be approximately matched to the sliding "window" of data signal to determine a match. Significantly, the same amount of data may be scanned much more quickly and data matching the search request may be determined much more quickly as well. For example, the inventors

25   have found that CPU based approximate searches of 200 megabytes of DNA sequences can take up to 10 seconds on a typical present day "high end" system, assuming the offline processing to index the database has already been completed. In that same 10 seconds, the inventors have found that a 10-gigabyte disk could be magnetically

30   searched for approximate matches using the present invention. This represents a 50:1 improvement in performance. Furthermore, in a typical hard disk drive there are four surfaces and corresponding read/write heads, which may be all searched in parallel should each head be equipped with the present invention. As these searches can

35   proceed in parallel, the total increase in speed or improvement represents a 200:1 advantage. Furthermore, additional hard disk drives may be accessed in parallel and scaled to further increase the advantage provided by the present invention.

7

By choosing an appropriate correlation or matching technique, and by setting an appropriate threshold, the search may be conducted to exactly match the desired signal, or more importantly and perhaps more powerfully, the threshold may be lowered to provide for

5   approximate matching searches. This is generally considered a more powerful search mode in that databases may be scanned to find "hits" which may be valid even though the data may be only approximately that which is being sought. This allows searching to find data that has been corrupted, incorrectly entered data, data which only

10   generally corresponds to a category, as well as other kinds of data searches that are highly desired in many applications. For example, a library of DNA sequences may be desired to be searched and hits found which represent an approximate match to a desired sequence of residues. This ensures that sequences which are close to the desired

15   sequence are found and not discarded but for the difference in a forgivable number of residue mismatches. Given the ever-increasing volume and type of information desired to be searched, more complex searching techniques are needed. This is especially true in the area of molecular biology, "[O]ne of the most powerful methods for

20   inferring the biological function of a gene (or the protein that it encodes) is by sequence similarity searching on protein and DNA sequence databases." Garfield, "The Importance of (Sub)sequence Comparison in Molecular Biology," pgs. 212-217, the disclosure of which is incorporated herein by reference. Current solutions for

25   sequence matching are only available in software or non-reconfigurable hardware.

Still another application involves Internet searches provided by Internet search engines. In such a search, approximate matching allows for misspelled words, differently spelled words, and other

30   variations to be accommodated without defeating a search or requiring a combinatorial number of specialized searches. This technique permits a search engine to provide a greater number of hits for any given search and ensure that a greater number of relevant web pages are found and cataloged in the search. Although, as mentioned above,

35   this approximate matching casts a wider net which produces a greater number of "hits" which itself creates its own problems.

Still another possible application for the present invention is for accessing databases which may be enormous in size or which may be stored as analog representations. For example, our society has seen

8

the implementation of sound recording devices and their use in many
forums including judicial proceedings. In recent history, tape
recordings made in the President's oval office have risen in
importance with respect to impeachment hearings. As can be
5   appreciated, tape recordings made over the years of a presidency can
accumulate into a huge database which might require a number of
persons to actually listen to them in order to find instances where
particular words are spoken that might be of interest. Utilizing the
present invention, an analog representation of that spoken word can
10   be used as a key and sought to be matched while the database is
scanned in a continuous manner and at rapid speed. Thus, the present
invention provides a powerful search tool for massive analog .
databases as well as massive digital databases.

        While text-based searches are accommodated by the present
15   invention as described above, storage media containing images, sound,
and other representations have traditionally been more difficult to
search than text. The present invention allows searching a large
data base for the presence of such content or fragments thereof. For
example, the key in this case could be a row or quadrant of pixels
20   that represent the image being sought. Approximate matching of the
key's signal can then allow identification of matches or near matches
to the key. In still another image application, differences in
pixels or groups of pixels can be searched and noted as results which
can be important for satellite imaging where comparisons between
25   images of the same geographic location are of interest as indicative
of movement of equipment or troops.

        The present invention may be embodied in any of several
configurations, as is noted more particularly below. However, one
important embodiment is perhaps in the form of a disk drive
30   accelerator which would be readily installed in any PC as an
interface between the hard disk drive and the system bus. This disk
drive accelerator could be provided with a set of standardized
templates and would provide a "plug and play" solution for
dramatically increasing the speed at which data could be accessed
35   from the drive by the CPU. This would be an after market or retrofit
device to be sold to the large installed base of PC's. It could also
be provided as part of a new disk drive, packaged within the envelope
of the drive case or enclosure for an external drive or provided as
an additional plug in pc card as an adapter for an internal drive.

9

Additional templates for various kinds of searches on various kinds of databases could be made available either with the purchase of the accelerator, such as by being encoded on a CD, or even over the Internet for download, as desired.

5   While the principal advantages and features of the present invention have been briefly explained above, a more thorough understanding of the invention may be attained by referring to the drawings and description of the preferred embodiment which follow.

Brief Description of the Drawings

10   Figure 1 is a block diagram illustrating an information search and retrieval system in accordance with one embodiment of the present invention;

        Figure 2 is a schematic of a conventional rigid disk drive system illustrating different insertion points for connection of the

15   present invention;

        Figure 3 is a block diagram of one embodiment of the transformation of a search inquiry processed by the system of Fig. 1;

        Figure 4 is a block diagram of one embodiment of a hardware implementation of the present invention used to conduct an exact

20   match search in a digital domain;

        Figure 5 is a block diagram of one embodiment of a hardware implementation of the present invention used to conduct an approximate match search in a digital domain;

        Figure 6 is a block diagram depicting the implementation of the

25   present invention in a stand-alone configuration;

        Figure 7 is a block diagram depicting the present invention implemented as a shared remote mass storage device across a network;

        Figure 8 is a block diagram depicting the present invention as a network attached storage device (NASD);

30   Figure 9 is a flowchart detailing the logical steps in the inventive method for searching and retrieving data from a magnetic storage medium;

        Figure 10 is a graphical representation of an analog signal as might be used as a data key;

35   Figure 11 is a graphical representation of an analog signal representing the continuous reading of data from a magnetic storage medium in which the data key is present;

        Figure 12 is a graphical representation of the signal of Fig. 10 overlying and matched to the signal of Fig. 11;

10

Figure 13 is a graphical representation of a correlation function calculated continuously as the target data in the magnetic storage medium is scanned and compared with the data key;

Figure 14 is a graphical representation of a correlation
5   function as the data key is continuously compared with a signal taken from reading a different set of target data from the magnetic storage medium but which also contains the data key;

Figure 15 is one embodiment of a table generated by the present invention for use in performing sequence matching operations;

10   Figure 16 is a block diagram of one embodiment of a systolic array architecture used by the present invention for computing the values of the table of Fig. 15;

Figures 17 and 18 are block diagrams of the systolic array architecture of Fig. 15 in operation during the combinational and
15   latch part of the clock cycle, respectively, of the system of Fig. 1;

Figure 19 is the table of Fig. 15 representing a particular sequence matching example;

Figure 20 is a block diagram of the systolic array architecture of Fig. 16 for the example of Fig. 19;

20   Figures 20, 21 and 22 are block diagrams of the systolic array architecture of Fig. 20 in operation during the combinational and latch part of the clock cycle, respectively, of the system of Fig. 1;

Figure 23 is a block diagram of one embodiment of a systolic array architecture used by the present invention in performing image
25   matching operations;

Figure 24 is a block diagram of another arrangement for the systolic array architecture in performing image matching operations;

Figure 25 is a block diagram of one embodiment of an individual cell of the systolic array shown in Fig. 23;

30   Figure 26 is a block diagram of another embodiment of an individual cell of the systolic array shown in Fig. 23;

Figure 27 is a block diagram showing an example using the present invention for performing data reduction operations; and

Figure 28 is a block diagram showing a more complex arrangement
35   of FPGA's.

Detailed Description of the Preferred Embodiment

As shown in Figure 1, the present invention is readily implemented in a stand-alone computer or computer system. In broad terms, the present invention is comprised of at least one re-

11

configurable logic device 21 coupled to at least one magnetic mass
storage medium 26, with that re-configurable logic device being an
FPGA. As depicted in Figure 1, the re-configurable logic device 21
may itself include a plurality of functional logic elements including

5   a data shift register and possibly a microprocessor, or they could be
on separate chips, or the individual logic elements could be
configured in a pipeline or parallel orientation as shown in some of
the other figures herein. In any event, re-configurable logic refers
to any logic technology whose form and function can be significantly

10   altered (i.e., reconfigured) in the field post-manufacture. Examples
of re-configurable logic devices include without limitation
programmable logic devices (PLDs). A PLD is an umbrella term for a
variety of chips that are programmable. There are generally three
physical structures for a PLD. The first is the permanent fuse type

15   which blows apart lines or fuses them together by electrically
melting an aluminum trace or insulator. This was the first type of
PLD, known as a "programmable array logic" or PAL. The second type
of PLD uses EEPROM or flash memory, and causes a transistor to open
or close depending on the contents of its associated memory cell.

20   The third type of PLD is RAM-based (which makes it dynamic and
volatile), and its contents are loaded each time it starts up. An
FPGA is an integrated circuit (IC) that contains an array of logic
units that can be interconnected in an arbitrary manner. These logic
units are referred to as CFB's or configurable logic blocks by one

25   vendor (Xilinx). Both the specific function of each logic unit and
the interconnections between logic units can be programmed in the
field after manufacture of the IC. FPGAs are one of the most common
PLD chips. FPGAs are available in all three structures. In the
preferred embodiment of the present invention, re-configurable logic

30   device 21 is constructed using Xilinx FPGA technology, and its
configuration is developed using the Mentor synthesis tools and the
Xilinx place-and-route tools, all of which are presently commercially
available as known to those of skill in the art.

The re-configurable logic device 21 interfaces with the system

35   or input/output bus 34 and, in one configuration, also interfaces
with any disk caches 30 which may be present. It receives and
processes search requests or inquires from the CPU 32 or network
interface 36. Additionally, the device may aid in passing the

results of the inquiries to either or both the disk cache 30 and/or
the CPU 32 (by way of the bus 34).

5   The mass storage medium 26 provides the medium for storing
large amounts of information which will hereafter be referred to as
target data. The term "mass storage medium" should be understood as
meaning any magnetic device used to store large amounts of data, and
which is typically designated for use in a computer or computer
network. Examples include without limitation hard disk drives or
sub-units such as a single disk surface, and these systems may be
10   rotating, linear, serial, parallel, or various combinations of each.
For example, a rack of hard disk drive units could be connected in
parallel and their parallel output provided at the transducer level
to one or more re-configurable logic devices 21. Similarly, a bank
of magnetic tape drives could be used, and their serial outputs each
15   provided in parallel to one or more re-configurable logic devices 21.
The data stored on the medium may be in analog or in digital form.
For example, the data could be voice recordings. The present
invention is thus scalable, permitting an increase in the amount of
data stored by increasing the number of parallel mass storage media,
20   while preserving the performance by increasing the number of parallel
re-configurable logic devices or replicating the re-configurable
logic device.

In the prior art as shown in the upper portion of Figure 1,
typically a disk controller 28 and/or a disk cache 30 may be used in
25   the traditional sense for access by a CPU 32 over its system or
input/output bus 34. The re-configurable logic device 21 accesses
target data in the mass storage medium 26 via one or more data shift
registers 24 and presents it for use at the system bus 34 without
moving large blocks of memory from the mass storage medium 26 over
30   the system bus 34 and into the working memory 33 of CPU 32 for
sorting and accessing. In other words, as is explained in greater
detail below, the CPU 32 may send a search request or inquiry to the
re-configurable logic device 21 which then asynchronously accesses
and sorts target data in the mass storage medium 26 and presents it
35   for use either in a disk cache 30 as is known in the prior art or
directly onto the system bus 34 without further processing being
required by CPU 32 or use of its working memory 33. The CPU 32 is
thus free to perform other tasks while the searching and matching
activity is being performed by the present invention. Alternately,

13

the control microprocessor may provide the search inquiry and template or programming instructions for the FPGA 21, and then perform the search and present the data on system bus 34 for access and use by CPU 32.

5   As has been explained above, the present invention may be used to perform a variety of different types of matching or data reduction operations on the target data. Each one of these operations will now be discussed in detail below. For all operations, however, it will be assumed that the target data is written onto the magnetic mass

10   storage medium 26 with sufficient formatting information attached so that the logical structure of the target data can be extracted. Exact and approximate string matching will be described with reference to Figures 2-5. It can be appreciated, however, that the present invention is not limited to single string matches and is

15   equally suitable for compound query matching (i.e., queries involving a plurality of text strings having a certain logical relationship therebetween or which use Boolean algebra logic). When performing an exact match with the re-configurable logic device 21 in the analog domain, shown as Point A in Figure 2, where matching is done using

20   analog comparators and correlation techniques, an exact match corresponds to setting a sufficiently high threshold value for matching the data key with analog target data on the mass storage medium 26. Approximate matching in the analog domain corresponds to setting appropriate (lesser) threshold values. The success of an

25   approximate match may be determined by the correlation value set in the re-configurable logic device 21 or by using one of a number of matching-performance metrics stored therein such as the number of bits within a data key that are equal to the corresponding bits in the scanned target data.

30   More particularly, a conventional rigid disk drive may have a plurality of rotating disks with multiple transducers accessing each disk. Each of these transducers typically has its output feeding analog signal circuitry 18, such as amplifiers. This is represented at point A. As further shown in Figure 2, typically the outputs of

35   the analog circuitry are selectively provided to a single digital decoder 23 which then processes one such output. This is represented at point B. This digital output is typically then sent through error correction circuitry (ECC) 25 and at its output C is then passed on to the bus 34 or disk cache 30. For purposes of the present

14

invention, it may be desirable to provide multiple parallel paths for
target data by providing multiple digital decoders and ECC's. Exact
matching in the digital domain could be performed at Point B or Point
C, which corresponds to the pre- and post-error-corrected digital

5   signal, respectively.

The results may be sent to a control microprocessor 22, which
may or may not be configured as part of an FPGA, to execute logic
associated with a compound or complex search inquiry. In the most
general case, a compound search inquiry 40 will go through the

10   transformation process illustrated in Figure 3. In particular, the
software system (not shown) that resides on the CPU 32 generates the
search inquiry 40. This inquiry proceeds through a compiler 42, also
located on the CPU 32, that is responsible for analyzing the search
inquiry. There are three main results from this analysis: (1)

15   determining the data key that will reside in the compare registers
within the re-configurable logic device 21; (2) determining the
combining logic that must be implemented in the control
microprocessor 22; and (3) producing hardware description 44 in a
standard hardware description language (HDL) format (or if possible

20   retrieving one from a library) that will be used to generate
synthesis commands 46 to the re-configurable logic device 21. Any
commercially available HDL and associated compiler and synthesis
tools may be used. The resulting logic functions may correspond to
exact or inexact matches or wildcard operations and simple word level

25   logic operations such as "and" and "or." This synthesis information
is sent to the control microprocessor 22 which acts to set up the re-
configurable logic device 21, or FPGA. In the case of complex logic
operations, a high-level language 48 such as C or C+ is used in
conjunction with a compiler 50 to generate the appropriate synthesis

30   commands to the microprocessor 22.

While the path shown in Figure 3 is able to handle a wide range
of potential search inquiries, it has the drawback that the latency
introduced into the search process might be too long. If the time
required for a search inquiry to flow through the transformations

35   represented in Figure 3 is of the same order as the time required to
perform a search, the compilation process might become the
performance bottleneck rather than the search itself. This issue can
be addressed for a wide range of likely search inquiries by
maintaining a set of precompiled hardware templates that handle the

most common cases. These templates may be provided and maintained
either in CPU 32 memory, made available through an off-line storage
medium such as a CD, or even kept in the mass storage medium 26
itself. Still further, such templates may be communicated to CPU 32
5   such as over a network or the Internet.

One embodiment of such a hardware template 29 is illustrated in
Figure 4. In particular, the data shift register 27 contains target
data streaming off the head (not shown) of one or more disks 19. A
compare register stores the data key for which the user wishes to
10   match. In the example shown, the data key is "Bagdad." Fine-grained
comparison logic device 31 performs element by element comparisons
between the elements of the data shift register 27 and the compare
register 35. The fine-grained comparison logic device 31 can be
configured to be either case sensitive or case insensitive. Word-
15   level comparison logic 37 is responsible for determining whether or
not a match at the world-level occurs. In the case of a compound
search inquiry, the word-level match signals are delivered to the
control microprocessor 22 for evaluation thereof. A match to the
compound search inquiry is then reported to the CPU 32 for further
20   processing.

One embodiment of a hardware template for conducting
approximate matching is illustrated in Figure 5. In particular, the
data shift register 27' contains target data streaming off the head
(not shown) of one or more disks 19'. A compare register 35' stores
25   the data key for which the user wishes to match. In the example
shown, the data key is again "Bagdad." Fine-grained comparison logic
31' performs element by element comparisons between the elements of
the data shift register 27' and the compare register 21'. Again, the
fine-grained comparison logic device 31' can be configured to be
30   either case sensitive or case insensitive.   The template 29'
provides for alternate routing of elements in data shift register 27'
to individual cells of the fine-grained comparison logic device 21'.
Specifically, each cell of the fine-grained comparison logic device
31' can match more than one position in the data shift register 27'
35   such that the compare register 21' can match both the commonly used
spelling of "Baghdad" as well as the alternate "Bagdad" in shared
hardware. Word-level comparison logic 37' is responsible for
determining whether or not a match at the word level occurs. In the

16

case of a compound search inquiry, the word-level match signals are delivered to the control microprocessor 22 for evaluation thereof. A match to the compound search inquiry is then reported to the CPU 32 for further processing.

5   The actual configuration of the hardware template will of course vary with the search inquiry type. By providing a small amount of flexibility in the hardware templates (e.g., the target data stored in the compare registers, the routing of signals from the data shift registers and compare register elements to the cells of

10   the fine-grained comparison logic device, and the width of the word-level comparison logic), such a template can support a wide range of word matches. As a result, this diminishes the frequency with which the full search inquiry transformation represented in Figure 3 must take place, which in turn, increases the speed of the search.

15   It should be noted that the data entries identified in an "approximate" match search will include the "exact" hits that would result from an "exact" search. For clarity, when the word "match" is used, it should be understood that it includes a search or a data result found through either of an approximate search or an exact

20   search. When the phrase "approximate match" or even just "approximate" is used, it should be understood that it could be either of the two searches described above as approximate searches, or for that matter any other kind of "fuzzy" search that has a big enough net to gather target data that are loosely related to the

25   search inquiry or in particular, data key. Of course, an exact match is just that, and does not include any result other than an exact match of the search inquiry with a high degree of correlation.

     Also shown in Figure 1 is a network interface 36 interconnecting the present invention to a network 38 which may be a

30   LAN, WAN, Internet, etc. and to which other computer systems 40 may be connected. With this arrangement, other computer systems 40 may conveniently also access the data stored on the mass storage medium 26 through the present invention 21. More specific examples are given below. Still further as shown in Figure 1, the elements 20-24

35   may themselves be packaged together and form a disk drive accelerator that may be separately provided as a retrofit device for adapting existing pc's having their own disk drives with the advantages of the present invention. Alternately, the disk drive accelerator may also be offered as an option on a hard drive and packaged in the same

17

enclosure for an external drive or provided as a separate pc board
with connector interface for an internal drive. Still further
alternatively, the disk drive accelerator may be offered as an option
by pc suppliers as part of a pc ordered by a consumer, business or

5   other end user. Still another embodiment could be that of being
offered as part of a larger magnetic mass storage medium, or as an
upgrade or retrofit kit for those applications or existing
installations where the increased data handling capability could be
used to good advantage.

10   As shown in Figures 6-8, the present invention may be
implemented in a variety of computer and network configurations. As
shown in Figure 6, the present invention may be provided as part of a
stand-alone computer system 41 comprising a CPU 43 connected to a
system bus 45 which then accesses a mass storage medium 47 having the

15   invention as disclosed herein.

           As shown in Figure 7, the mass storage medium 51 coupled with
the present invention may be itself connected directly to a network
52 over which a plurality of independent computers or CPU's 54 may
then access the mass storage medium 51. The mass storage medium 51

20   may itself be comprised of a bank of hard disk drives comprising a
RAID, disk farm, or some other massively parallel memory device
configuration to provide access and approximate matching capabilities
to enormous amounts of data at significantly reduced access times.

           As shown in Figure 8, a mass storage medium 56 coupled with the

25   present invention may be connected to a network 58 as a network
attached storage device (NASD) such that over the network 58 a
plurality of stand-alone computers 60 may have access thereto. With
such a configuration, it is contemplated that each mass storage
medium, represented for illustrative purposes only as a disk 57,

30   would be accessible from any processor connected to the network. One
such configuration would include assigning a unique IP address or
other network address to each mass storage medium.

           The configurations as exemplified by those shown in Figures 1
and 6-8 represent only examples of the various computer and network

35   configurations with which the present invention would be compatible
and highly useful. Others would be apparent to those having skill in
the art and the present invention is not intended to be limited
through the examples as shown herein which are meant to be instead
illustrative of the versatility of the present invention.

18

As shown in Figure 9, the method of the present invention for use in exact or approximate matching is described alternatively with respect to whether an analog or digital data domain is being searched. However, beginning at the start of the method, a CPU

5   performs certain functions during which it may choose to access target data stored in a mass storage medium. Typically, the CPU runs a search inquiry application 62 which may be representative of a DNA search, an Internet search, an analog voice search, a fingerprint search, an image search, or some other such search during which an

10   exact or approximate match to target data is desired. The search inquiry contains directives specifying various parameters which the disk control unit 28 and the re-configurable logic device 20 must have to properly obtain the data key from the mass storage medium 26. Examples of parameters include but are not limited to the following:

15   the starting location for scanning the storage device; the final location after which (if there is not match) scanning is terminated; the data key to be used in the scanning; a specification of the approximate nature of the matching; and what information should be returned when a match occurs. The sort of information that can be

20   returned includes the address of the information where the match was found, or a sector, record, portion of record or other data aggregate which contains the matched information. The data aggregate may also be dynamically specified in that the data returned on a match may be specified to be between bounding data specifiers with the matched

25   data contained within the bounding field. As the example in Figure 5 shows, looking for the word "bagdad" in a string of text might find the approximate match, due to misspelling, of the word "Baghdad", and return a data field which is defined by the surrounding sentence. Another query parameter would indicate whether the returned

30   information should be sent to the system or input/output bus 34, or the disk cache 30.

Referring back to Figure 9, the search inquiry will typically result in the execution of one or more operating system utilities. As an example of a higher level utility command, for the UNIX

35   operating system, this could be modified versions of glimpse, find, grep, apropos, etc. These functions cause the CPU to send commands 66 such as search, approximate search, etc., to the re-configurable logic device 21 with relevant portions of these commands also being sent to the disk controller 28 to, for example, initiate any mass

storage medium positioning activity 69 that is later required for properly reading target data from the mass storage medium.

At this point, depending upon the particular methodology desired to be implemented in the particular embodiment of the

5   invention, it would be necessary that an analog or digital data key is determined. This data key, which can be either exact or approximate for a text search, corresponds to the data being searched for. For an analog data key, it may either be pre-stored such as in the mass storage medium, developed using dedicated circuitry, or

10   required to be generated. Should the analog data key be pre-stored, a send pre-stored data key step 68 would be performed by the microprocessor 22 (see Fig. 1) which would transmit the data key in digital and sampled format to the re-configurable logic device 20 as shown in step 70. Alternatively, should the analog data key not be

15   pre-stored, it can be developed using one of a number of mechanisms, two of which are shown in Figure 9. In one, the microprocessor 22 would write the data key on the magnetic mass storage medium as at step 72 and then next read the data key as at step 74 in order to generate an analog signal representation of the data key. In

20   another, as at step 71, the digital version of the data key received from the CPU would be converted using appropriate digital to analog circuitry to an analog signal representation which would in turn be appropriately sampled. The data key would then next be stored as a digital sample thereof as in step 70. Should a digital data key be

25   used, it is only necessary that the microprocessor 22 store the digital data key as at step 76 in the compare register of the re-configurable logic device. It should be understood that depending upon the particular structures desired to be included for each re-configurable logic device, the data key may reside in either or all

30   of these components, it merely being preferable to ultimately get the appropriate digital format for the data key into the re-configurable logic device 21 for comparison and correlation.

Next, after the mass storage medium 26 reaches its starting location as at 79, the target data stored on the mass storage medium

35   is continuously read as at step 78 to generate a continuous stream signal representative of the target data. Should an analog data key have been used, this analog data key may then be correlated with an analog read of the target data from the mass storage medium 26 as at step 80.

20

While the inventors contemplate that any of many prior art comparators and correlation circuitry could be used, for present purposes the inventors suggest that a digital sampling of the analog signal and data key could be quite useful for performing such
5   comparison and calculating the correlation coefficient, as explained below. It is noted that this analog signal generated from reading the target data from mass storage medium 26 may be conveniently generated by devices in the prior art from the reading of either analog or digital data, it not being necessary that a digital data
10   key be used to match digital target data as stored in mass storage medium 26. Alternatively, a correlation step 82 may be performed by matching the digital data key with a stream of digital target data as read from the mass storage medium 26. It should be noted that the data key may reflect the inclusion of approximate information or the
15   re-configurable logic device 21 may be programmed to allow for same. Thus, correlating this with target data read from the mass storage medium enables approximate matching capabilities.

Referring back to Figure 9, decision logic 84 next makes an intelligent decision as to whether a portion of the target data
20   approximately matches or does not approximately match the data key. Should a match be found, then the target data is processed as at step 86 and the key data requested by the search inquiry is sent to a disk cache 30, directly onto system bus 34, or otherwise buffered or made available to a CPU 32, network interface 36, or otherwise as shown in
25   Figures 1, and 6-8. A logical step 88 is preferably included for returning to the continuous reading of target data from the mass storage medium 26, indicating something like a "do" loop. However, it should be understood that this is a continuous process and that target data is processed from the mass storage medium 26 as a stream
30   and not in individualized chunks, frames, bytes, or other predetermined portions of data. While this is not precluded, the present invention preferably allows a data key to be in essence "slid" over a continuously varying target data read signal such that there is no hesitation in reading target data from the mass storage
35   medium 26. There is no requirement to synchronize reading to the start or end of any multi-bit data structure, or any other intermediate steps required to be performed as the target data is compared continuously "on the fly" as it is read from the mass

21

storage medium 26. Eventually, the data access is completed as at step 90 and the process completed.

The inventors herein have preliminarily tested the present invention in the analog domain and have generated preliminary data demonstrate its operability and effectiveness. In particular, Figure 10 is a graphical representation of a measured analog signal output from a read/write head as the read/write head reads a magnetic medium on which is stored a 10-bit digital data key. As shown therein, there are peaks in the analog signal which, as known in the art, represents the true analog signal generated by a read/write head as target data is read from a magnetic medium such as a hard disk. The scales shown in Figure 10 are volts along the vertical axis and tenths of microseconds along the horizontal axis. As shown in Figure 11, an analog signal is generated, again by a read/write head, as target data is read from a pseudo-random binary sequence stored in a test portion of a magnetic medium. The read signal does not provide an ideal square wave output when examined at this level.

Figure 12 is a graphical representation, with the horizontal scale expanded, to more specifically illustrate the overlap between approximately two bits of the 8-bit data key and the corresponding two bits of target data found in the pseudo-random binary sequence encoded at a different location on the disk or magnetic medium.

Figure 13 is a graphical representation of a correlation coefficient calculated continuously as the comparison is made between the data key and the continuous reading of target data from the hard disk. This correlation coefficient is calculated by sampling the analog signals at a high rate and using prior art signal processing correlation techniques. One such example may be found in Spatial Noise Phenomena of Longitudinal Magnetic Recording Media by Hoinville, Indeck and Muller, IEEE Transactions on Magnetics, Volume 28, no. 6, November 1992, the disclosure of which is incorporated herein by reference. A prior example of a reading, comparison, and coefficient calculation method and apparatus may be found in one or more of one of the co-inventor's prior patents, such as US Patent No. 5,740,244, the disclosure of which is incorporated herein by reference. The foregoing represent examples of devices and methods which may be used to implement the present invention, however, as mentioned elsewhere herein, other similar devices and methods may be likewise used and the purposes of the invention fulfilled.

22

As shown in Figure 13, at approximately the point labeled 325, a distinct peak is noted at approximately 200 microseconds which approaches 1 Volt, indicating a very close match between the data key and the target data. Figure 10 is also illustrative of the

5   opportunity for approximate matching which is believed to be a powerful aspect of the present invention. Looking closely at Figure 13, it is noted that there are other lesser peaks that appear in the correlation coefficient. Thus, if a threshold of 0.4 Volts were established as a decision point, then not only the peak occurring

10   which approaches 1 would indicate a match or "hit" but also another five peaks would be indicative of a "hit". In this manner, a desired coefficient value may be adjusted or predetermined as desired to suit particular search parameters. For example, when searching for a particular word in a large body of text, lower correlation values may

15   indicate the word is present but misspelled.

Figure 14 depicts the continuous calculation of a correlation coefficient between the same 8-bit data key but with a different target data set. Again, a single match is picked up at approximately 200 microseconds where the peak approaches 1 Volt. It is also noted

20   that should a lower threshold be established additional hits would also be located in the target data.

As previously mentioned, the present invention is also capable of performing sequence matching searches. With reference to Figure 15, a table 38 is generated by the re-configurable logic device 20 to

25   conduct such a search. Specifically, p_1 p_2 p_3 p_4 represents the data key, p, or desired sequence to be searched. While the data key of Figure 15 only shows four characters, this is for illustrative purposes only and it should be appreciated that a typical data key size for sequence searching is on the order of 500-1000, or even

30   higher. The symbols t_1, t_2, $t_3...t_9$ represent the target data, t, streaming off of the mass storage medium 26. Again, while only nine (9) characters of such data are shown, it should be appreciated that the typical size of the mass storage medium 26 and thus the target data streaming off of it can typically be in the range of several

35   billion characters. The symbols $d_{i,j}$ represent the edit distance at position i in the data key and position j in the target data. It is assumed that the data key is shorter relative to the target data, although it is not required to be so. There may be a set of known

(constant) values for an additional row (d0,j) and column (di,0) not
shown in Figure 15.

The values for di,j are computed by the re-configurable logic
device 20 using the fact that di,j is only a function of the

5   following characters: (1) pi, (2) tj, (3) di-1,j-1, (4) di-1,j, and
(5) di,j-1. This is illustrated in Figure 15 with respect to the
position d3,6 by showing its dependency on the values of d2,5 and
d2,6 and d3,5 as well as p3 and t6. In one embodiment, the values
for di,j are computed as follows:

10 di,j = max[di,j-1 + A; di-1,j + A; di-1,j-1 + Bi,j],
where A is a constant and Bi,j is a tabular function of pi and tj.
The form of the function, however, can be quite arbitrary. In the
biological literature, B is referred to as the scoring function. In
the popular database searching program BLAST, scores are only a

15 function of whether or not pi = tj. In other contexts, such as for
amino acid sequences, the value of B is dependent upon the specific
characters in p and t.

Figure 16 shows one embodiment of a systolic array architecture
used by the present invention to compute the values in the table 38

20 of Figure 15. The characters of the data key are stored in the
column of data registers 53, while the characters of the target data
streaming off of the mass storage medium 26 are stored in the data
shift registers 55. The values of di,j are stored in the systolic
cells 59 which themselves are preferably FPGA's.

25 The operation of the array of Figure 16 will now be illustrated
using Figures 17 and 18. As shown in Figure 17, in the first (i.e.,
combinational) part of the clock cycle of the system, the four
underlined values are computed. For example, the new value d3,6 is
shown to depend upon the same five values illustrated earlier in

30 Figure 15. As shown in Figure 18, in the second (i.e., latch) part
of the clock cycle, all the characters in di,j and tj are shifted one
position to the right. A comparator 61 is positioned at each
diagonal cell of the d array and determines when the threshold has
been exceeded.

35 The sequence matching operation will now be described with
reference to Figures 19-22 with respect to the following example:
key = axbacs
target data = pqraxabcstvq
A = 1

24

> B = 2, if i = j
>
> B = -2 if i = j

From these variables, the table of Figure 19 is generated by the re-
configurable logic device 20. Assuming a pre-determined threshold of
"8", the re-configurable logic device 20 will recognize a match at
d6,9.

A portion of the synthesis arrays representing the values
present in Figures 16-18 for this example are shown in Figures 20-22,
respectively. A match is identified by the re-configurable logic
device 20 when the value on any row exceeds a predetermined
threshold. The threshold is set based on the desired degree of
similarity desired between the data key and the target data stored in
mass memory device 26. For example, in the case of an exact match
search, the data key and target data must be identical. The match is
then examined by the CPU 32 via a traceback operation with the table
of Figure 19. Specifically a "snapshot" of the table is sent to the
CPU 32 at a predetermined time interval to assist in traceback
operations once a match is identified. The interval is preferably
not too often to overburden the CPU 32, but not so infrequent that it
takes a lot of time and processing to recreate the table. To enable
the CPU 32 to perform the traceback operation, it must be able to
recreate the d array in the area surrounding the entry in the table
that exceeded the threshold. To support this requirement, the
systolic array can periodically output the values of a complete
column of d ("a snapshot") to the CPU 32. This will enable the CPU
32 to recreate any required portion of d greater than the index j of
the snapshot.

Many matching applications operate on data representing a two
dimensional entity, such as an image. Figure 23 illustrates a
systolic array 120 of re-configurable logic devices 20, preferably
FPGA's, which enables matches on two dimensional data. The
individual cells 122 each hold one pixel of the image for which the
user is desiring to match (the image key) and one pixel of the image
being searched (the target image). For images of sufficiently large
size, it is likely they will not all fit into one re-configurable
logic chip 124. In such cases, a candidate partitioning of cells to
chips is shown with the dashed lines, placing a rectangular subarray
of cells in each chip 124. The number of chip-to-chip connections

25

can be minimized by using a subarray that is square (i.e., same
number of cells in the vertical and horizontal dimension). Other
more complicated arrangements are shown below.

Loading of the target image into the array 120 is explained
5   using Figure 24. Individual rows of each target image streaming off
the mass magnetic medium 26, shown generally as point A, into the top
row 130 of the array via the horizontal links 134 connecting each
cell. With such a configuration, the top row 130 operates as a data
shift register. When the entire row 130 is loaded, the row is
10   shifted down to the next row 132 via the vertical links 136 shown in
each column. Once the entire image is loaded into the array, a
comparison operation is performed, which might require arbitrary
communication between neighboring cells. This is supported by both
the horizontal and vertical bi-directional links 126 and 128,
15   respectively, shown in Figure 23.

Although for simplicity purposes the individual bi-directional
links 126 and 128 are shown simply in Figures 23 and 24, Figure 28
shows the flexibility for implementing a much more complex set of bi-
directional links. As shown in Figure 28, data may be communicated
20   from a mass storage medium 180 and be input to a first row of a
plurality of cells 182, with each cell of the first row having a
direct link to the corresponding cell 184 below it in a second row of
cells with a simple link 186, and so on throughout the array 188 of
cells. Overlying the array 188 of cells is a connector web 190 which
25   provides direct connectivity between any two cells within the array
without the need for transmission through any intervening cell. The
output of the array 188 is represented by the sum of the exit links
192 at the bottom of the array 188. It should be understood that
each cell in the array may be comprised of an FPGA, each one of which
30   preferably has a re-configurable logic element corresponding to
element 20 in Figure 1, or any one of which may have a re-
configurable logic element 20 as well as a data shift register 24, or
any one of which may have the entirety of re-configurable logic
device 21.

35   One embodiment for the individual cells of array 120 is
illustrated in Figure 25. The cell 140 includes a pixel register
142, LOADTi,j, which contains the pixels of the target image
currently being loaded into the array. A register, 144 CMPTi,j,
contains a copy of the pixel register 142 once the complete target

26

image has been loaded. This configuration enables the last target
image loaded to be compared in parallel with the next target image
being loaded, essentially establishing a pipelined sequence of load,
compare, load, compare, etc. A register 146, CMPPi,j, contains the

5 pixels of the image key to be used for comparison purposes, and the
compare logic 148 performs the matching operation between register
144 and register 146. The compare logic 148 may include the ability
to communicate with the neighboring cells to the left, right, up, and
down shown generally as 150, 152, 154, and 156, respectively, to

10 allow for complex matching functions.

Another embodiment for the individual cells of array 120 of
Figure 23 is illustrated in Figure 26. The cell 140 of Figure 25 has
been augmented to support simultaneous loading of the image key and
the target image. In particular, the cell 160 includes the same

15 components of the cell 140, but adds a new register 162, LOADPi,j,
which is used to load the image key, and is operated in the same
manner as register 142. With such a configuration, if one disk read
head of the mass storage medium 26 is positioned above the image key,
and a second disk read head is positioned above the target image,

20 they can both flow off the disk in parallel and be concurrently
loaded into the array 160.

The operation performed within the compare logic block can be
any function that provides a judgment as to whether or not there are
significant differences between the target image and the image key.

25 An example includes cross-correlations across the entire image or
sub-regions of the image as described in John C. Russ, *The Image
Processing Handbook*, 3rd edition, CRC Press 1999, which is
incorporated herein by reference.

The present invention is also capable of performing data

30 reduction searching. Such searching involves matching as previously
described herein, but includes summarizing the matched data in some
aggregate form. For example, in the financial industry, one might
want to search financial information to identify a minimum, maximum,
and latest price of a stock. A re-configurable logic device for

35 computing such aggregate data reductions is illustrated as 100 in
Figure 27. Here, a data shift register 102 reads target data from a
mass storage medium containing stock price information. In the
example shown, three data reduction searches are shown, namely
calculating the minimum price, the maximum price, and the latest

27

price. As target data is fed into the data shift register 102, decision logic computes the desired data reduction operation. In particular, the stock price is fed to a minimum price comparator 110 and maximum price comparator 112 and stored therein. Each time a

5   stock price is fed to comparator 110, it compares the last stored stock price to the stock price currently being fed to it and whichever is lower is stored in data register 104. Likewise, each time a stock price is fed to comparator 112, it compares the last stored stock price to the stock price currently being fed to it and

10   whichever is higher is stored in data register 106. In order to compute the latest price, the stock price is fed into a data register 108 and the current time is fed into a comparator 114. Each time a time value is fed into comparator 114, it compares the last stored time with the current time and which ever is greater is stored in

15   data register 116. Then, at the end of the desired time interval for which a calculation is being made, the latest price is determined.

While data reduction searching has been described with respect to the very simple financial example shown in Figure 27, it can be appreciated that the present invention can perform data reduction

20   searching for a variety of different applications of varying complexity requiring such functionality. The re-configurable logic device need simply be configured with the hardware and/or software to perform the necessary functions

The ability to perform data reduction searching at disk

25   rotational speeds cannot be under-estimated. One of the most valuable aspects of information is its timeliness. People are growing to expect things at Internet speed. Companies that can quickly compute aggregate data reductions will clearly have a competitive advantage over those that cannot.

30   Various changes and modifications to the present invention would be apparent to those skilled in the art but yet which would not depart from the spirit of the invention. The preferred embodiment describes an implementation of the invention but this description is intended to be merely illustrative. Several alternatives have been

35   also been above. For example, all of the operations exemplified by the analog processing have their equivalent counterparts in the digital domain. Thus, approximate matching and correlation types of processing can be done on the standard digital representation of the analog bit patterns. This can also be achieved in a continuous

28

fashion using tailored digital logic, microprocessors and digital signal processors, or alternative combinations. It is therefore the inventors' intention that the present invention be limited solely by the scope of the claims appended hereto, and their legal equivalents.

5

What is claimed is:

1.   A data searching device for connection between a magnetic data
storage device and a CPU, said data searching device comprising a
FPGA, said FPGA being re-configurable to perform a search on data
stored in said data storage device.

2.   The data searching device of claim 1 wherein said CPU is
configured to transmit a template to said FPGA to thereby reconfigure
it.

3.   The data searching device of claim 2 wherein said CPU is
configured to transmit a key to said FPGA for matching to data read
from the data storage device in performing the search.

4.   The data searching device of claim 3 further comprising a data
shift register connected between the data storage device and the
FPGA, the data storage device being configured to store target data
as it is read from the data storage device.

5.   The data searching device of claim 1 wherein said device further
comprises a dedicated CPU connected between the FPGA and the other
CPU, said dedicated CPU being configured to transmit a template to
said FPGA to thereby reconfigure it.

6.   The data searching device of claim 5 wherein said dedicated CPU
is configured to transmit a key to said FPGA for matching to data
read from the data storage device in performing the search.

7.   The data searching device of claim 6 further comprising a data
shift register connected between the data storage device and the
FPGA, the data storage device being configured to store target data
as it is read from the data storage device.

8.   The data searching device of claim 1 wherein said FPGA comprises
a dedicated CPU and a re-configurable logic device.

9.   The data searching device of claim 8 wherein said FPGA further
comprises a data shift register.

30

10. A hard disk drive accelerator for connection between a hard disk drive and a processor, said accelerator comprising an FPGA having processing capability and arranged so that data read from the hard disk drive flows to the FPGA for processing prior to being passed on to the processor.

11. The hard disk drive accelerator of claim 10 further comprising a shift register connected to the FPGA and configured to receive data as it is read from the hard disk.

12. The hard disk drive accelerator of claim 9 further comprising a pc board, and wherein said hard disk drive accelerator is mounted thereto.

13. The hard disk drive accelerator of claim 9 wherein said FPGA comprises a data processor, a re-configurable logic device, and a data shift register.

14. A data storage and retrieval device comprising at least one mass data storage device configured to at least temporarily store target data and at least one re-configurable logic device comprising an FPGA coupled to the at least one mass data storage device, said re-configurable logic device being configurable to read a continuous stream of target data therefrom.

15. The data storage and retrieval device of claim 14 wherein the mass data storage device is a hard disk.

16. The data storage and retrieval device of claim 15 wherein the re-configurable logic device is configured to receive at least one search inquiry in the form of a data key, and to determine a match between the data key and the target data as it is read from the at least one mass data storage device.

17. The data storage and retrieval device of claim 16 wherein the re-configurable logic device is configured to search for matches between an analog data key and an analog target data.

18. The data storage and retrieval device of claim 16 wherein the
re-configurable logic device is configured to search for matches
between a digital data key and a digital target data.

19. The data storage and retrieval device of claim 14 further
comprising a plurality of magnetic storage media, each of said
magnetic storage media having an associated re-configurable logic
device.

20. The data storage and retrieval device of claim 14 wherein the
data key is a sequence having a plurality of elements, and the re-
configurable logic device comprises:

     a table generator configured to generate a table having a
5   plurality of table entries based on the elements of the sequence and
the target data read from the at least one magnetic storage medium;
and

     a comparator configured to compare each table entry to a
predetermined threshold and determine the match based on the
10   comparison therebetween.

21. The data storage and retrieval device of claim 14 wherein the
data key is an image having a plurality of pixels, and the re-
configurable logic device comprises an array generator configured to
generate at least one array having a plurality of cells, each cell
5   being configured to determine a match between one pixel of the data
key and one pixel of the target data as it is read from the at least
one magnetic storage medium.

22. The data storage and retrieval device of claim 21 wherein each
cell of the array comprises:

     a data key pixel register configured to store the data key
pixels;

5   a target data pixel register configured to load the target data
pixels; and

     a comparator configured to compare the data key pixels stored
in the data key pixel register with the target data pixels stored in
the target data pixel register to determine whether there is a match
10   therebetween.

23. The data storage and retrieval device of claim 22 wherein the comparator is further configured to communicate with other neighboring cells of the array to determine a match between each cell and its other neighboring cells.

24. The data storage and retrieval device of claim 22 further comprising a copy register configured to store a copy of the target data pixel register once an entire target image has been loaded therein and wherein the comparator is configured to compare in parallel the last image loaded in the target data pixel register with the current image being loaded in the target data pixel register.

25. The data storage and retrieval device of claim 22 further comprising a load data key pixel register configured to load the data key pixels of the array concurrently with the loading of the target data pixels in the target data pixel register.

26. The data storage and retrieval device of claim 14 wherein the re-configurable logic device is configured to compute aggregate data reductions on any matches determined thereby.

27. The data storage and retrieval device of claim 26 further comprising a plurality of re-configurable logic devices, and a plurality of data shift registers configured to store target data as it is read from the at least one magnetic storage medium, each data shift register connected between the at least one magnetic storage medium and each re-configurable logic device.

28. A device for retrieving data from at least one magnetic storage medium, the device being coupled to said at least one magnetic storage medium and interfacing the at least one magnetic storage medium with a computer network desiring the retrieved data for processing thereof, the device comprising a re-configurable logic device comprising an FPGA configured to compare a data key representative of the data sought to be retrieved with a target data representative of data read from the at least one magnetic storage medium.

29. The device of claim 28 wherein the computer network includes at least one computer, and wherein that one computer is configured to reconfigure the FPGA for different types of search protocols.

30. The device of claim 28 wherein the FPGA further comprises processor capability for reconfiguring the FPGA for different types of search protocols.

31. The device of claim 28 further comprising a data processor for reconfiguring the FPGA for different types of search protocols.

32. A network attached storage device, comprising:
    at least one magnetic storage medium; and
    at least one re-configurable logic device coupled to the at least one magnetic storage medium, the network attached device having
5   a network addressable input/output port configured to receive search inquiries and respond thereto.

33. A method of retrieving data from at least one magnetic storage medium, comprising:
    receiving a search inquiry in the form of a data key from a processor in communication with the at least one magnetic storage
5   medium;
    receiving target data from the at least one magnetic storage medium;
    comparing the data key with the target data as it is read from the at least one magnetic storage medium with a re-configurable logic
10 device comprising an FPGA; and
    determining a match based on the comparison therebetween.

34. The method of claim 33 wherein the data key is a sequence having a plurality of elements and wherein the method further comprises the steps of:
    generating a table having a plurality of table entries based on
5   the elements of the sequence and the target data read from the at least one magnetic storage device;
    comparing each table entry to a predetermined threshold; and
    determining a match based on the comparisons therebetween.

35. The method of claim 34 wherein the data key is an image having a plurality of data key pixels, the target data are images having a plurality of target data pixels, and wherein the method further comprises the steps of:

storing the data key pixels in a data key register;

loading the target data pixels into a target data pixel register;

comparing the data key pixels and the target data pixels; and

determining a match based on the comparisons therebetween.

36. The method of claim 35 further comprising generating at least one array having a plurality of cells, each cell configured to perform the storing, loading and comparing steps of claim 33.

37. The method of claim 35 further comprising:

storing a copy of the target data pixel register once an entire target image has been loaded therein; and

comparing in parallel the last image loaded in the target data pixel register with the current image being loaded in the target data pixel register.

38. The method of claim 37 further comprising loading the data key pixels concurrently with the loading of the target data pixels in the target data pixel register.

39. The method of claim 38 further comprising computing aggregate data reductions on any determined matches.
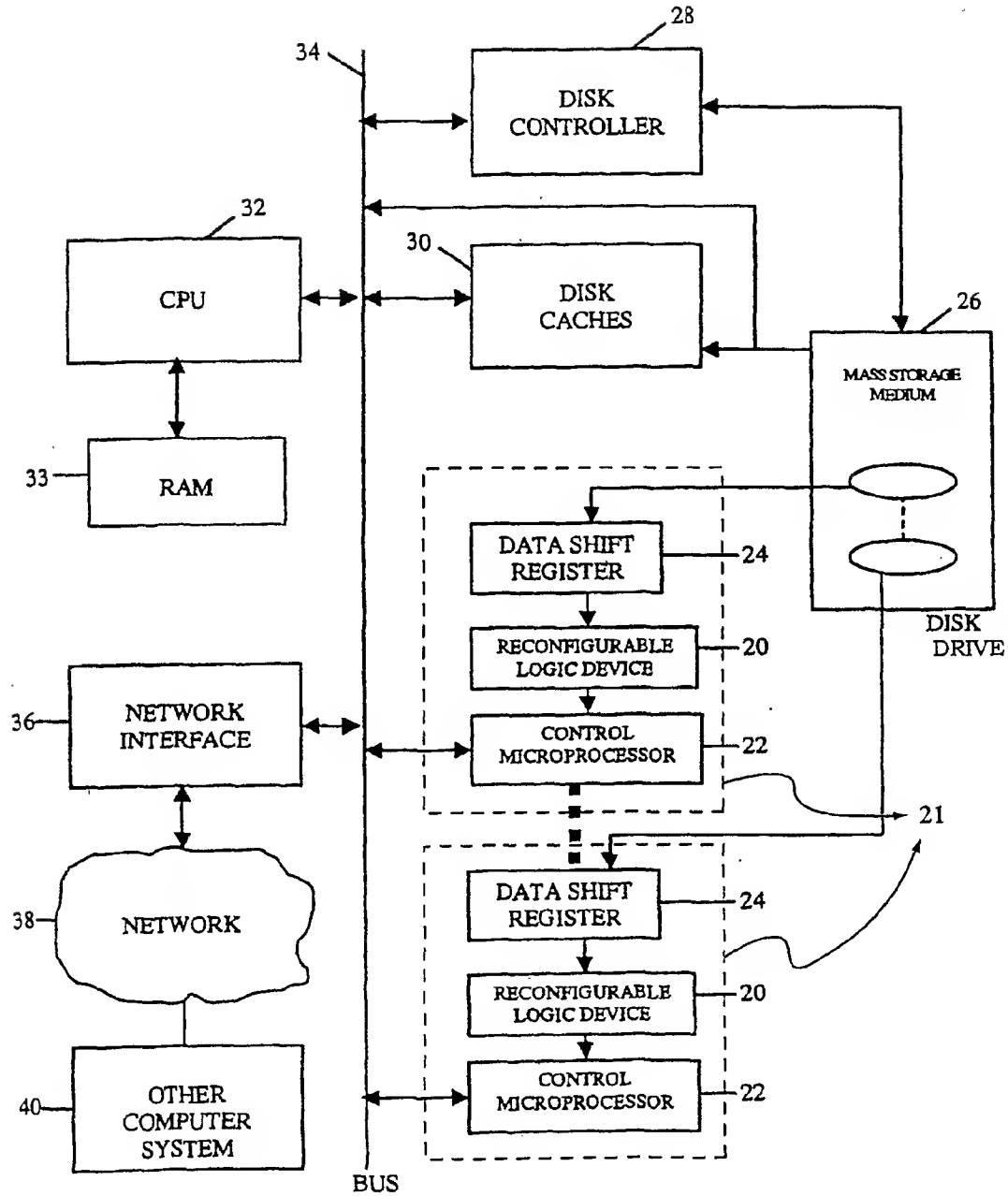
Figure 1
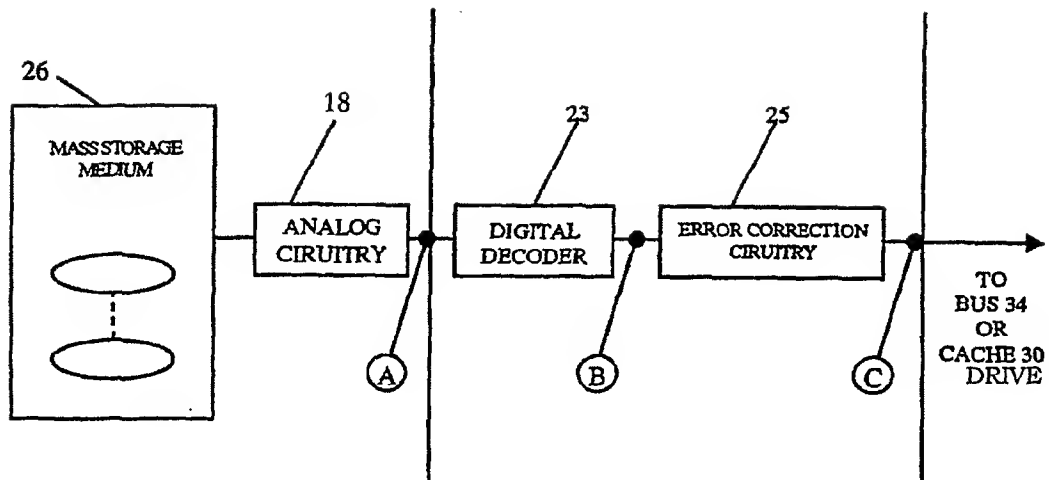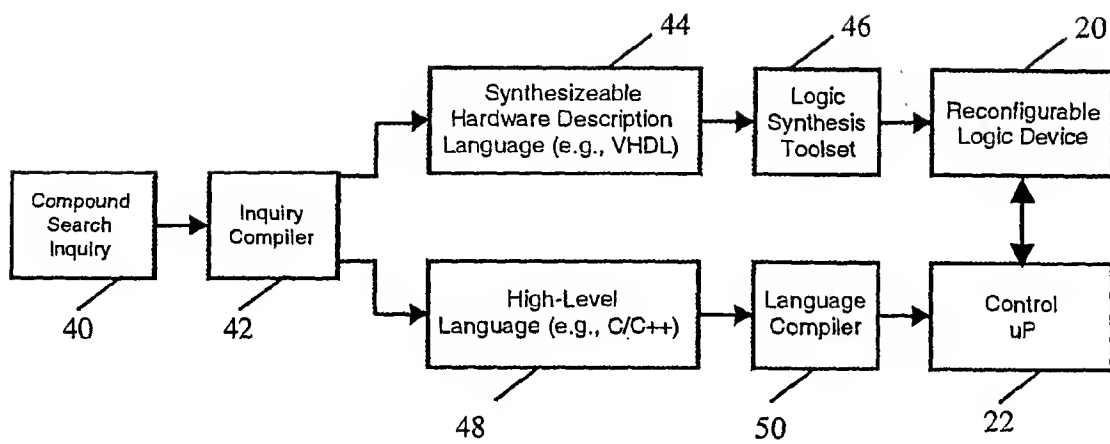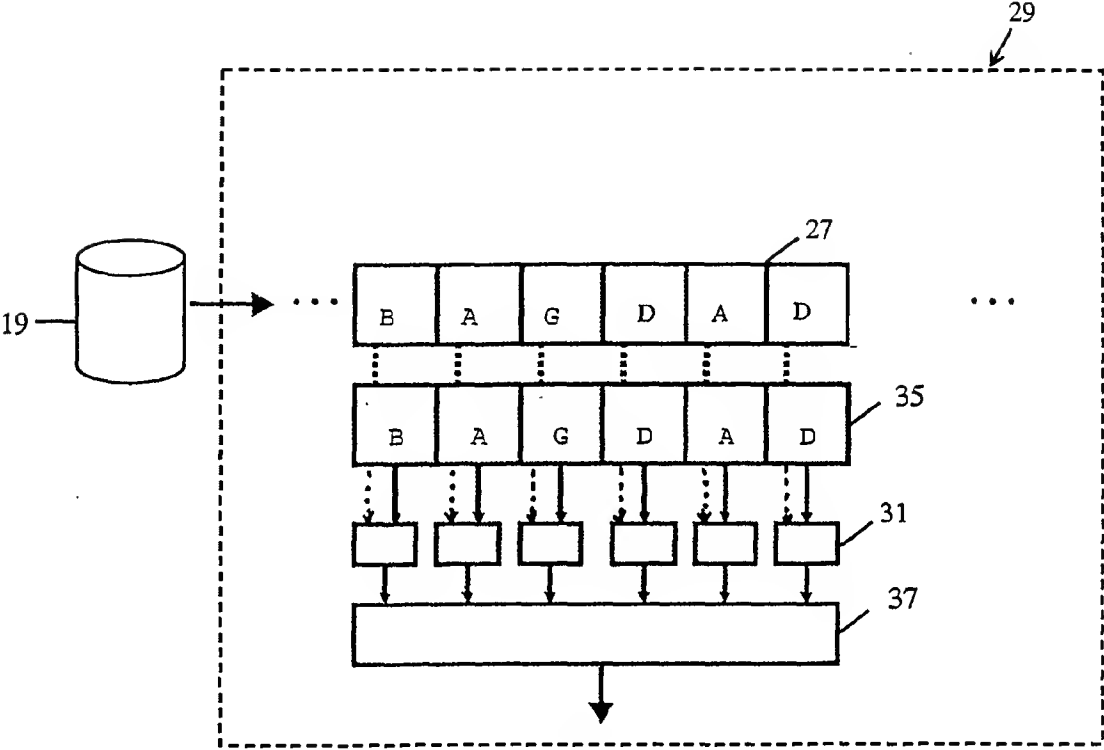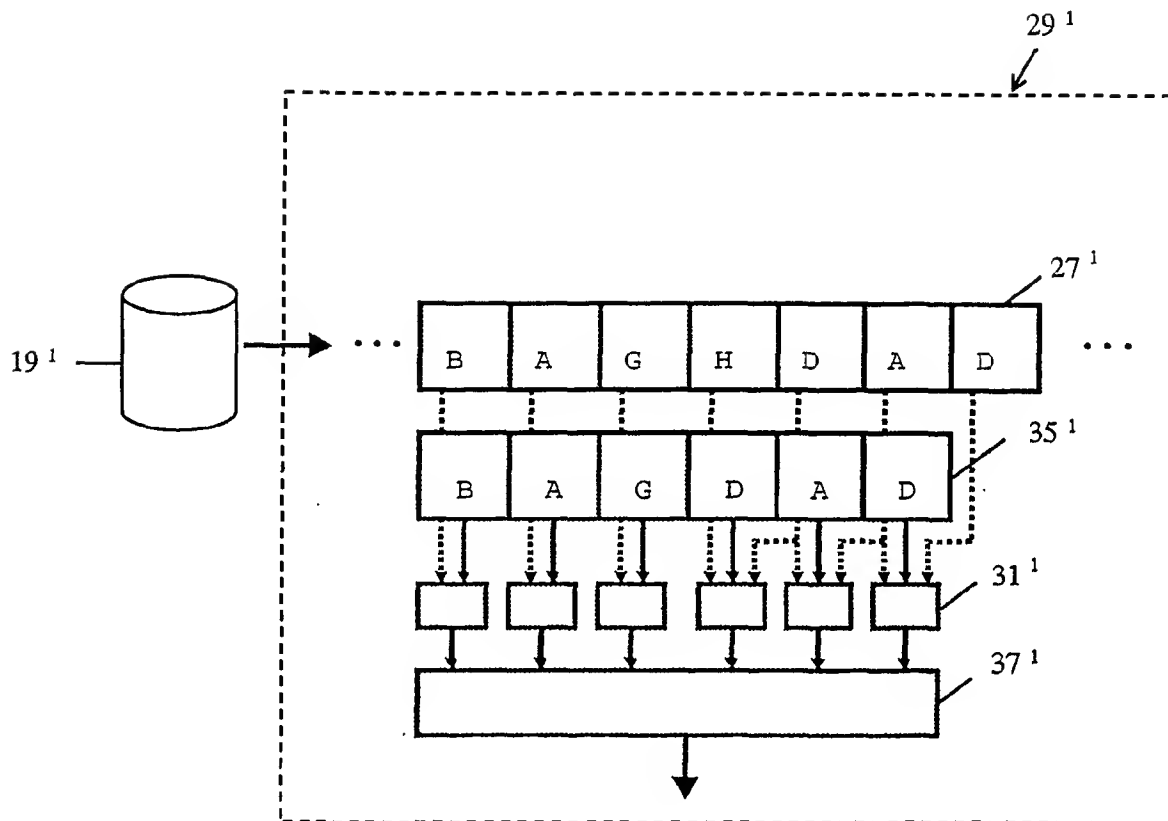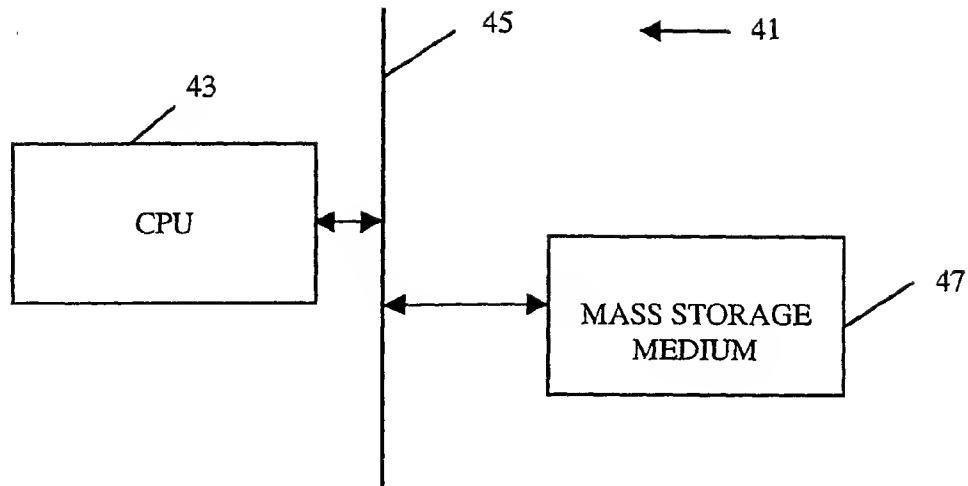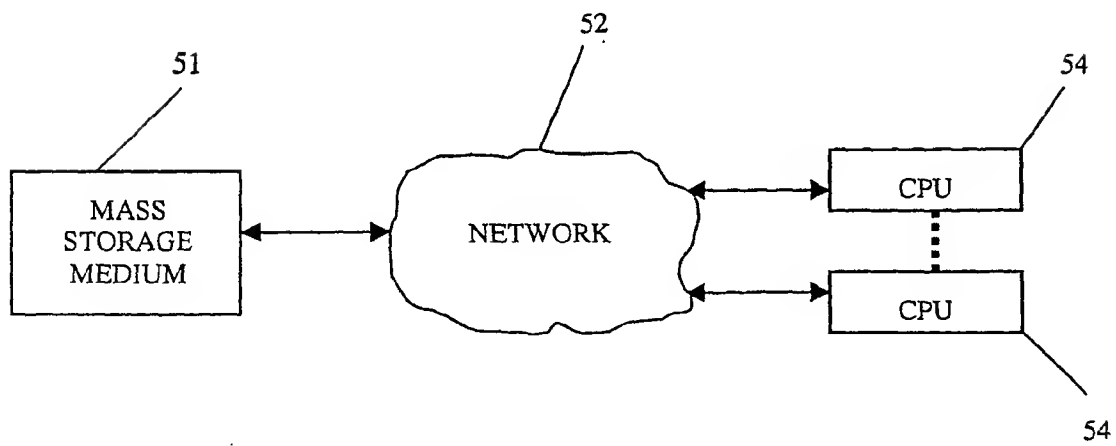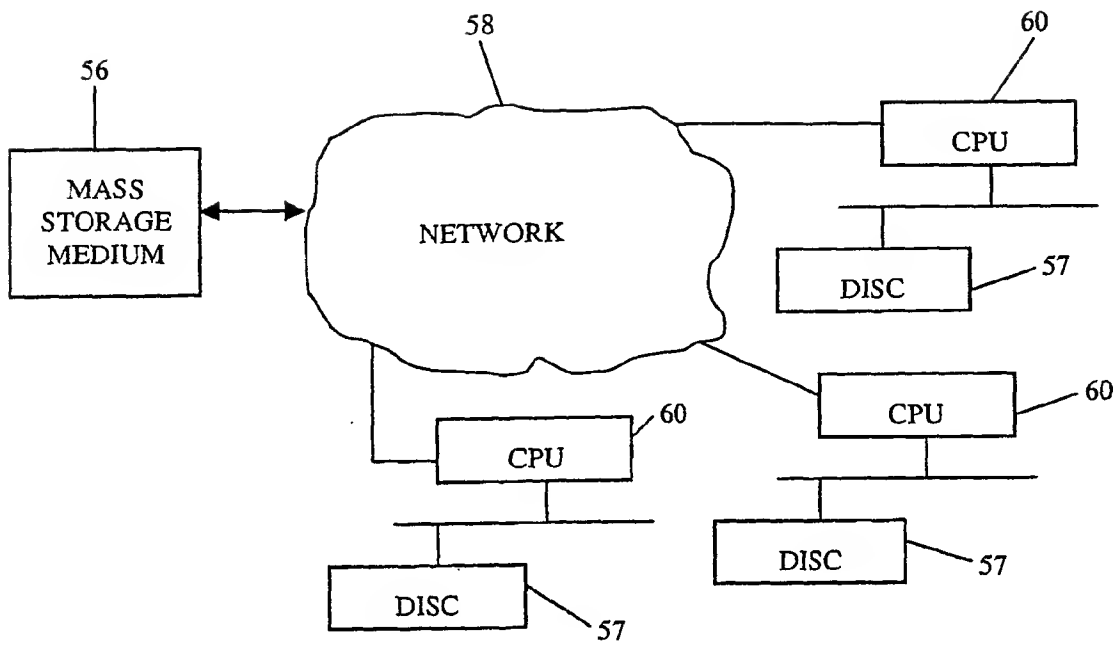
Figure 2

Figure 3

Figure 4

Figure 5

Figure 6

Figure 7

Figure 8

Figure 9

Figure 10

Figure 11

Figure 12

Figure 13

Figure 14

	t1	t2	t3	t4	t5	t6	t7	t8	t9
p1	d1,1	d1,2	d1,3	d1,4	d1,5	d1,6	d1,7	d1,8	d1,9
p2	d2,1	d2,2	d2,3	d2,4	d2,5	d2,6	d2,7	d2,8	d2,9
p3	d3,1	d3,2	d3,3	d3,4	d3,5	d3,6	d3,7	d3,8	d3,9
p4	d4,1	d4,2	d4,3	d4,4	d4,5	d4,6	d4,7	d4,8	d4,9

38
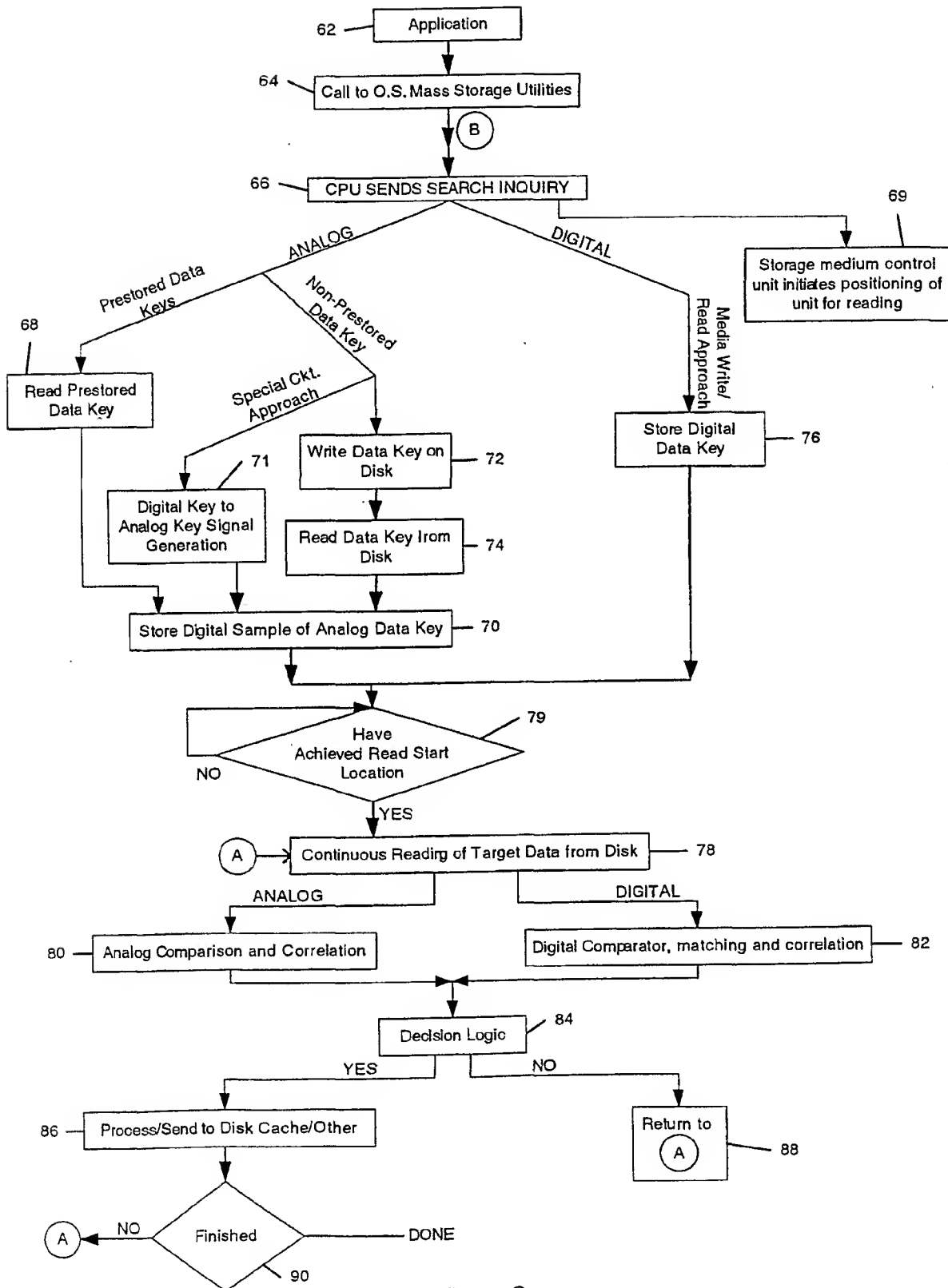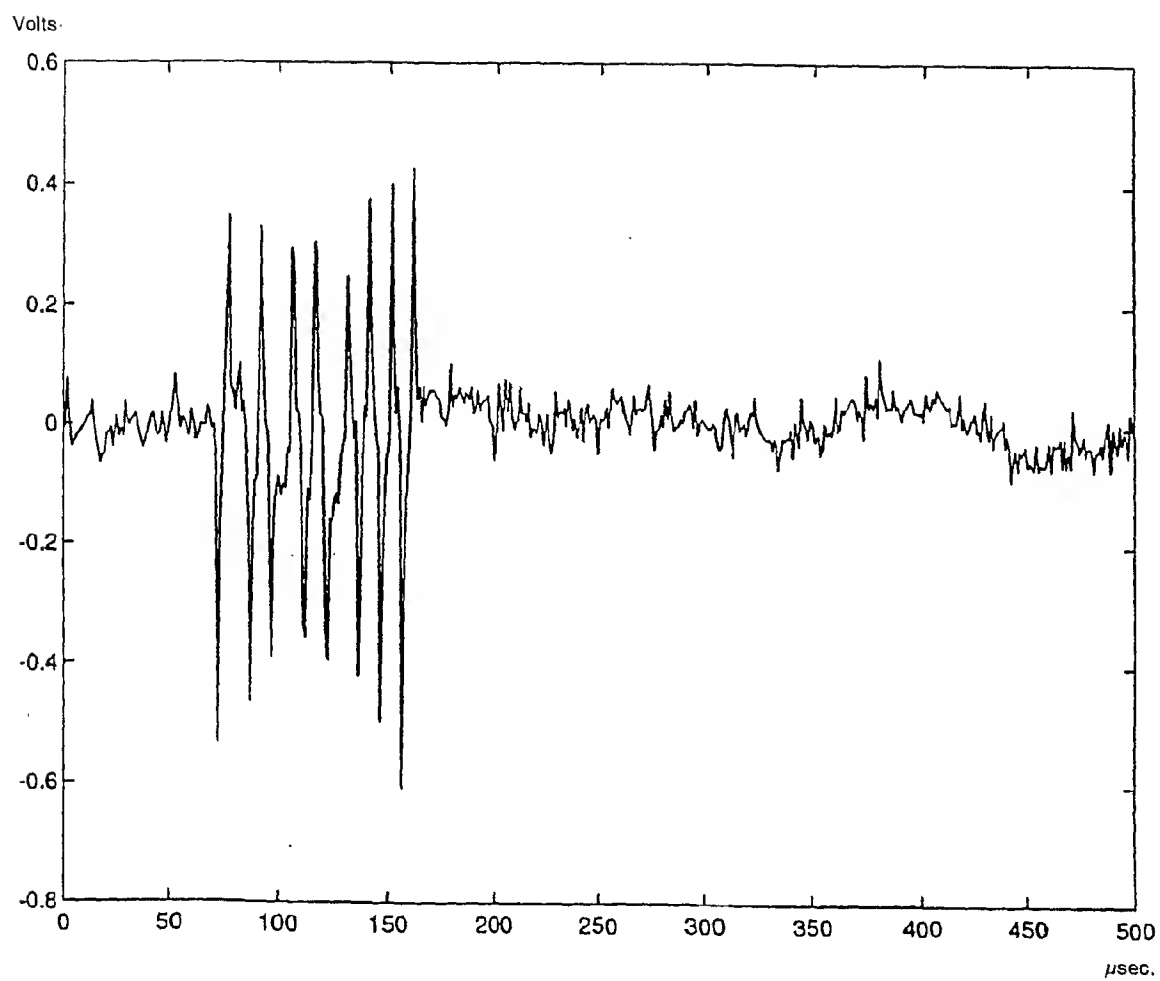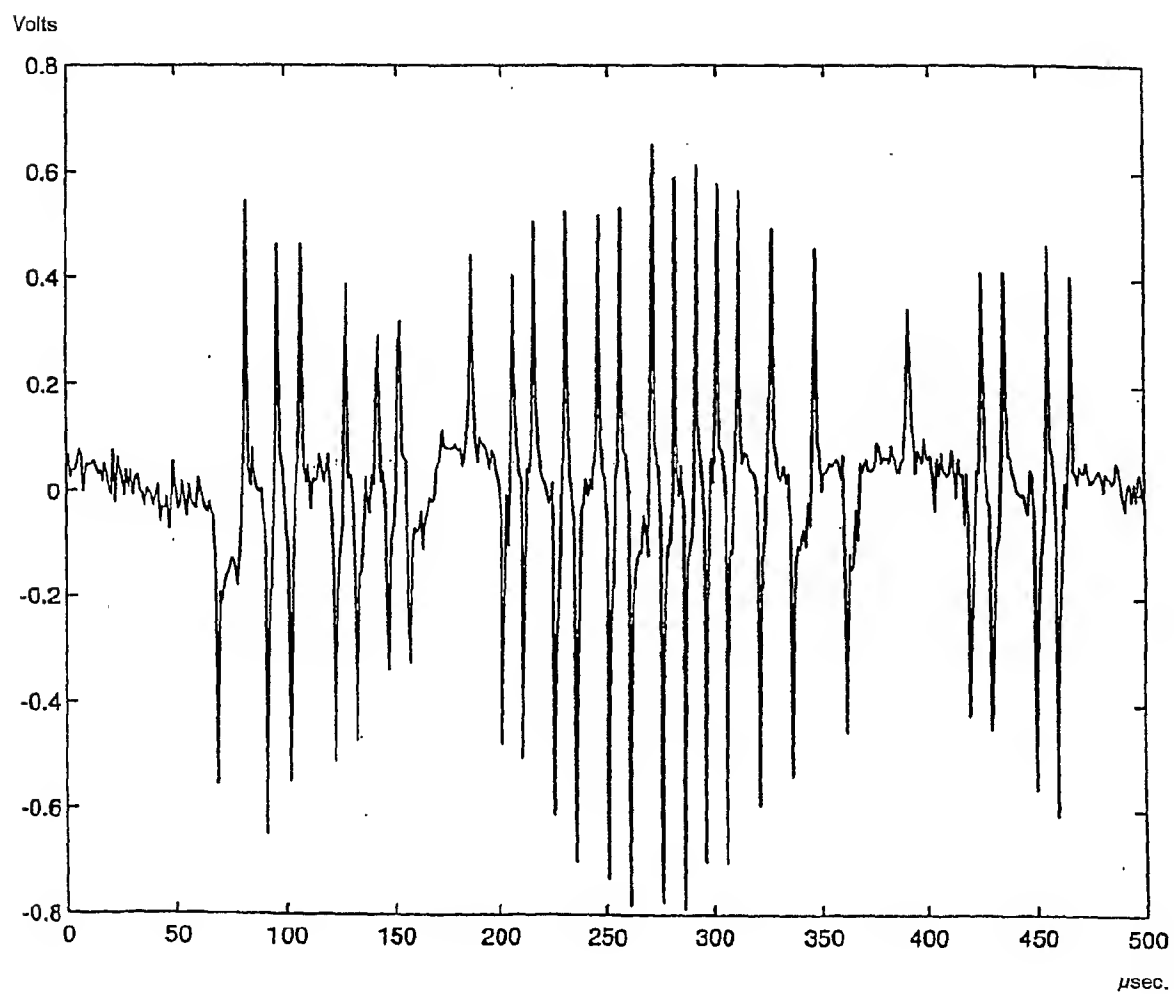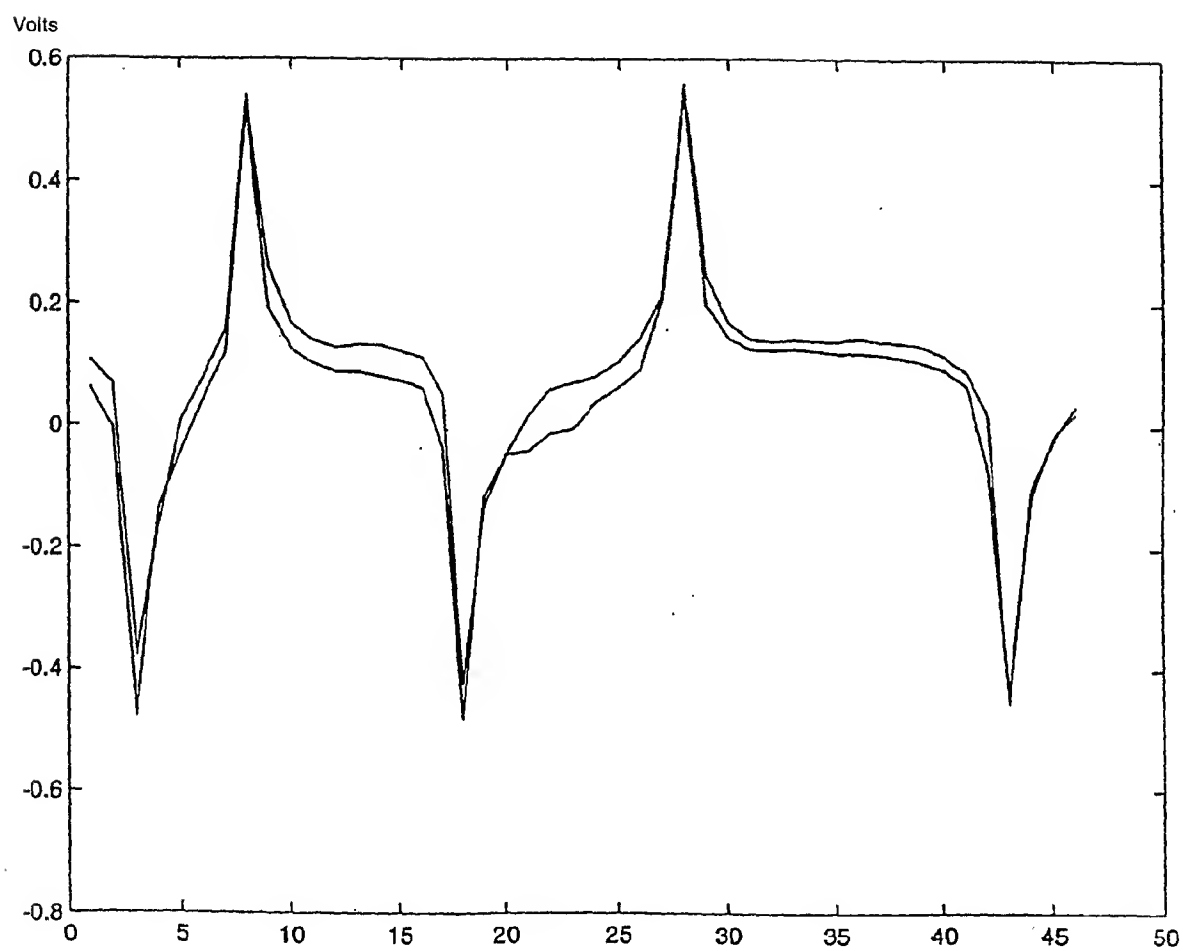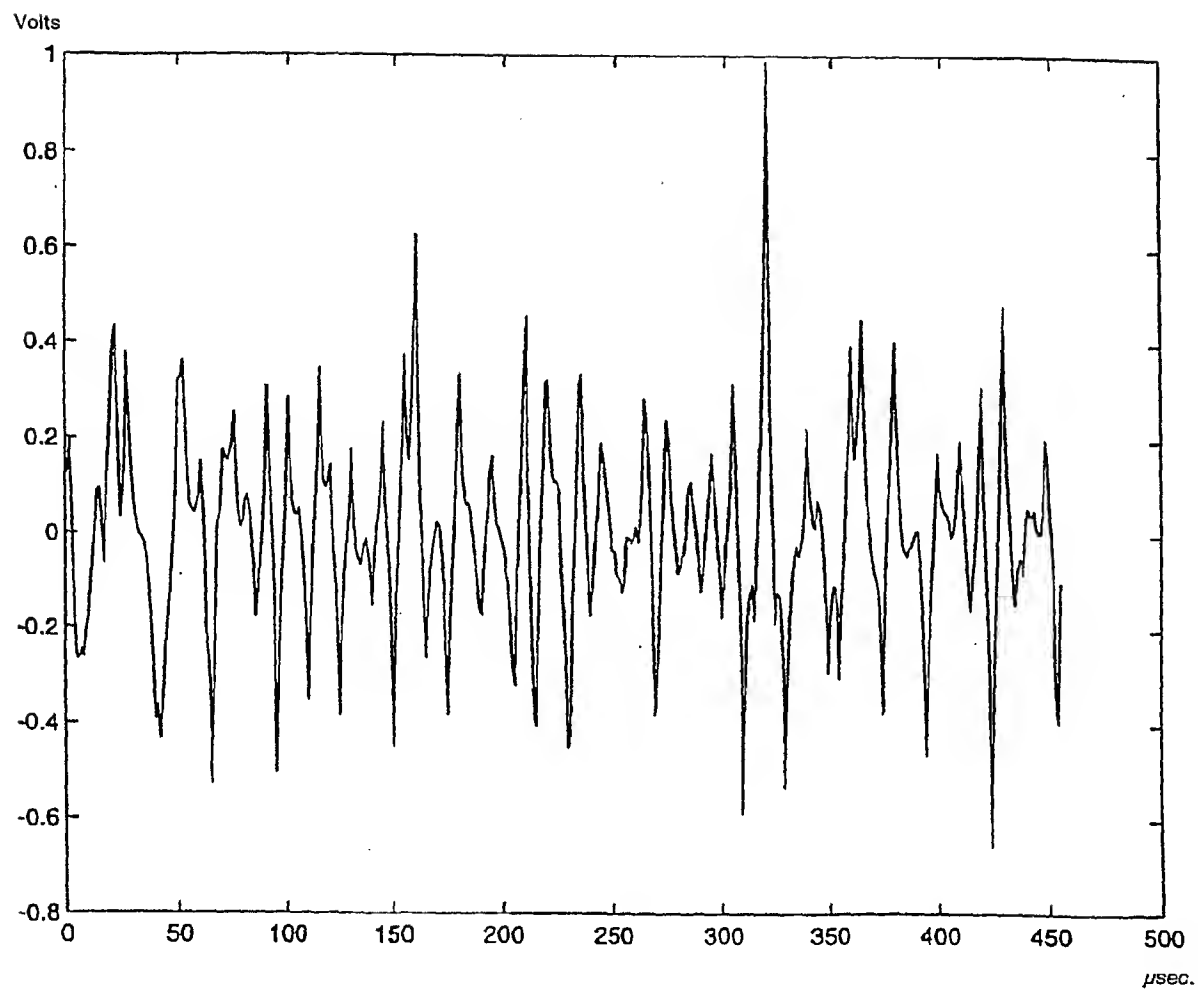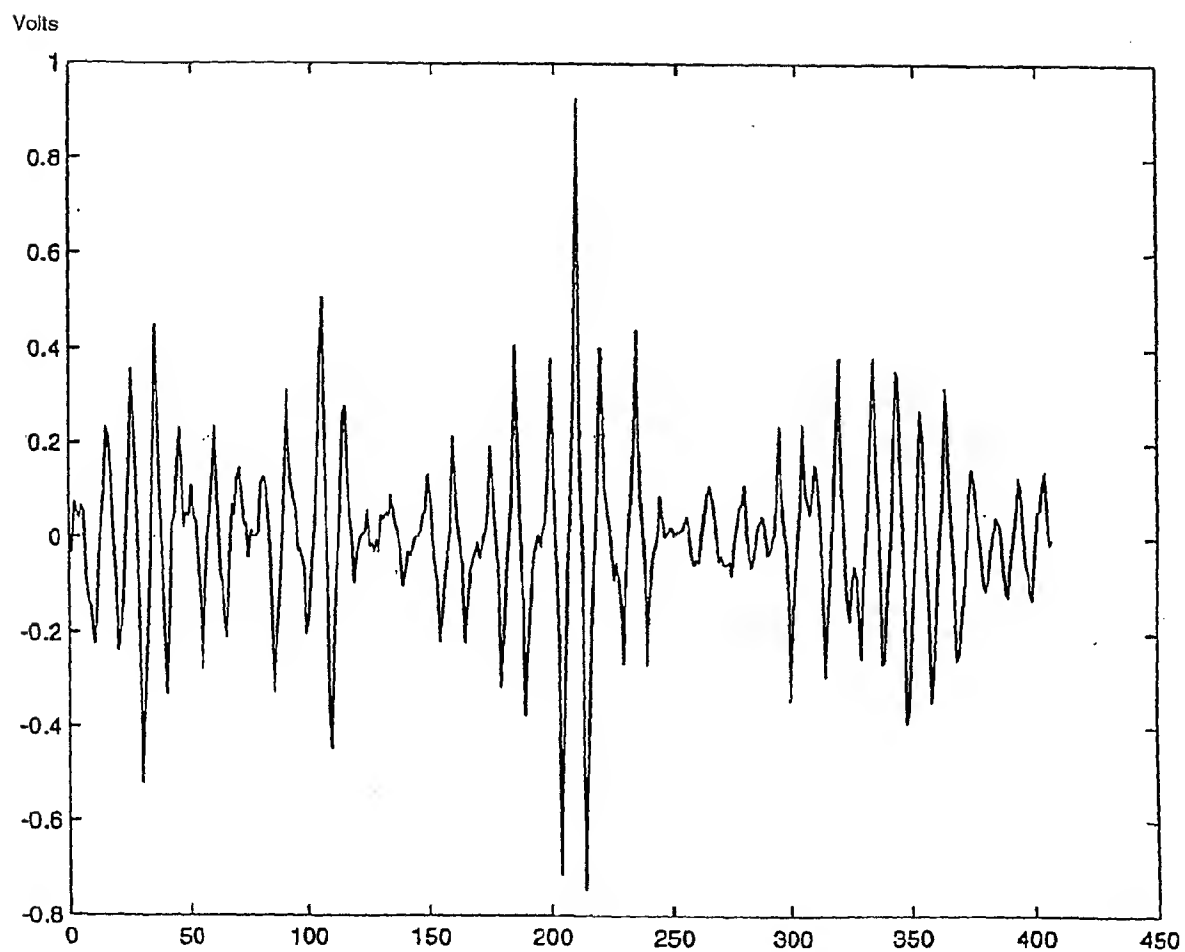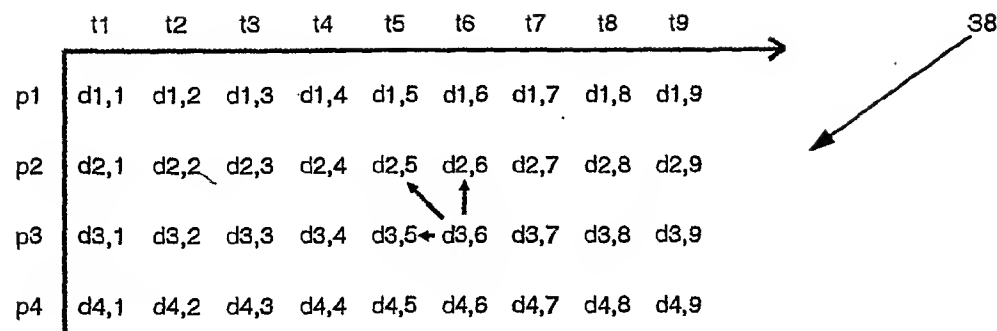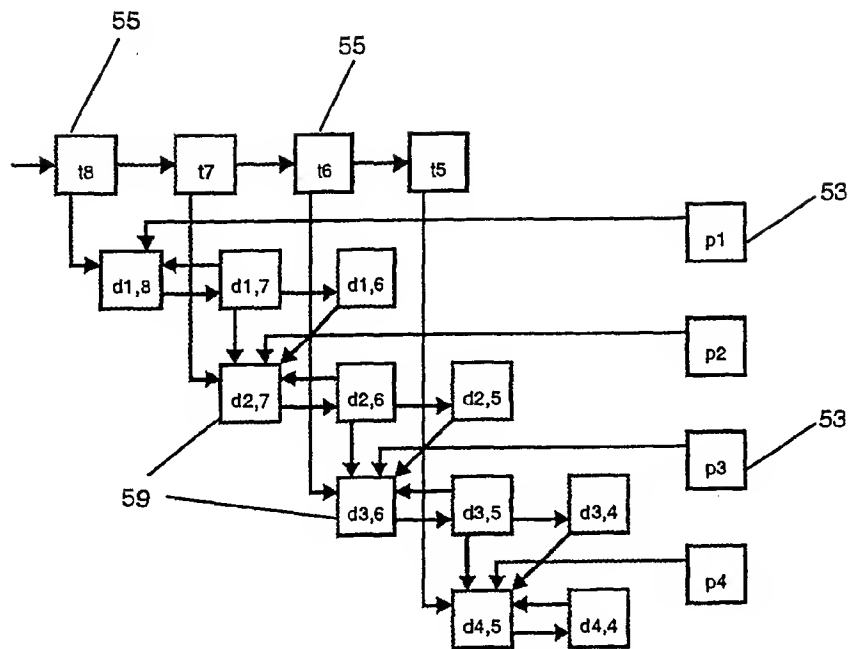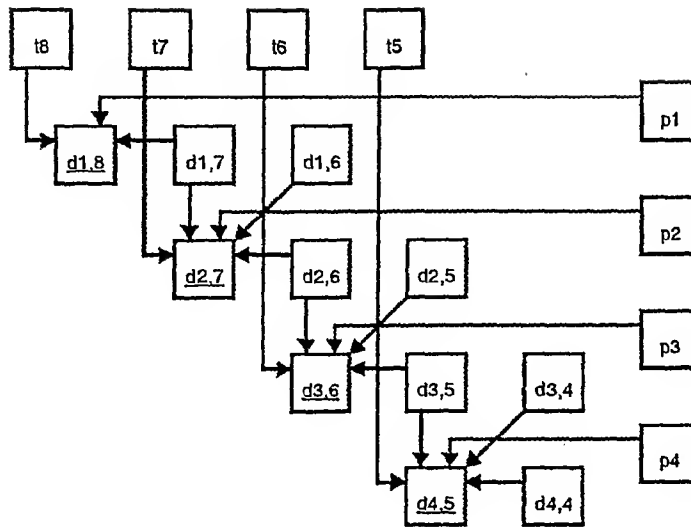
Figure 15
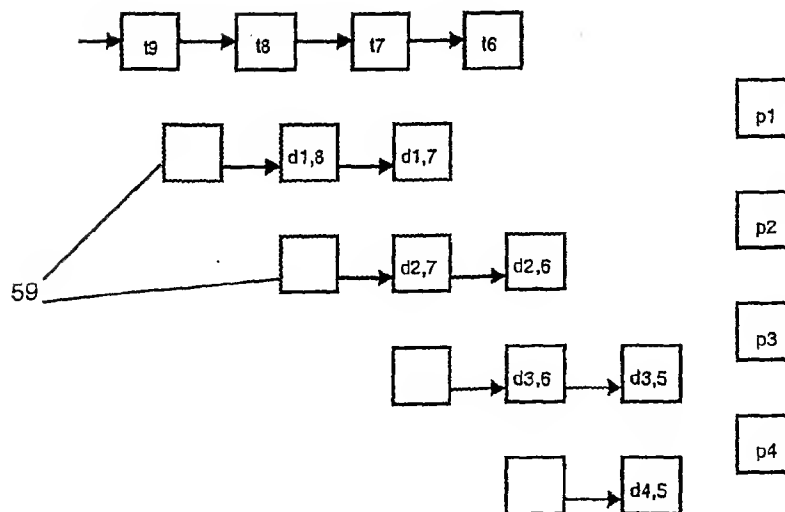
Figure 16

Figure 17

Figure 18

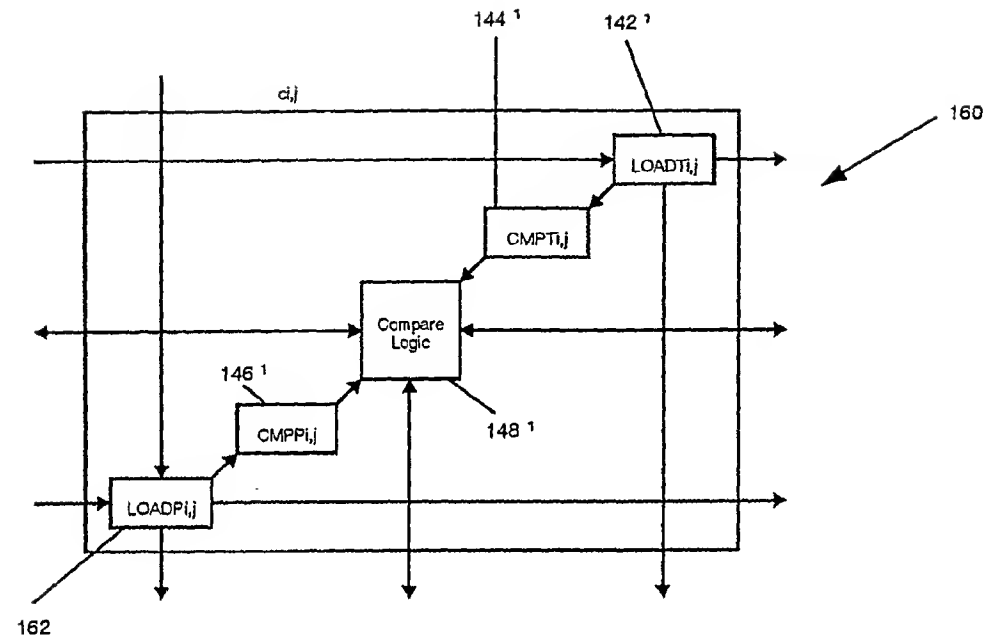|	p	q	r	a	x	a	b	c	s	t	v	q ...
a	-1	-1	-1	2	1	2	1	0	-1	-1	-1	-1
x	-1	-2	-2	1	4	3	2	1	0	-1	-2	-2
b	-1	-2	-3	0	3	2	5	4	3	2	1	0
a	-1	-2	-3	-1	2	1	4	3	2	1	0	-1
c	-1	-2	-3	-2	1	0	3	6	5	4	3	2
s	-1	-2	-3	-3	0	-1	2	5	_8_	7	6	5

Figure 19
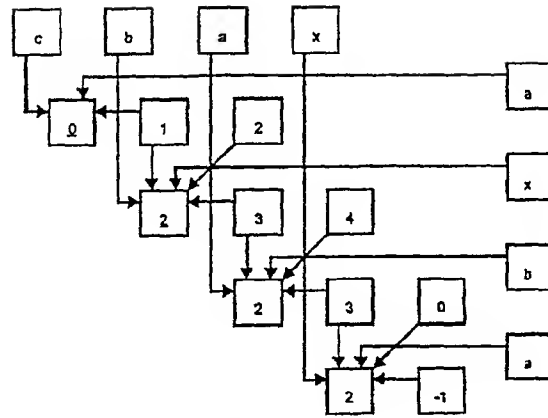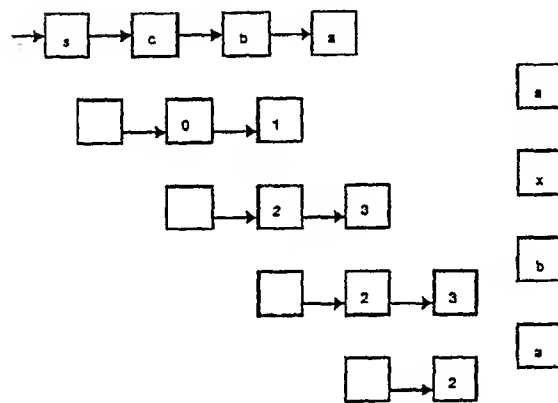
Figure 20

Figure 21

Figure 22

Figure 23

Figure 24

Figure 25

Figure 26

Figure 27

Fig. 28